Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Avaliação Externa de Modelos de Tópicos por Word Embedding na Língua Portuguesa

Alex S. Lacerda

Monografia apresentada como requisito parcial
para conclusão do Curso de Engenharia de Computação

Orientador
Prof. Dr. Thiago de Paulo Faleiros

Brasília
2019

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Avaliação Externa de Modelos de Tópicos por Word Embedding na Língua Portuguesa

Alex S. Lacerda

Monografia apresentada como requisito parcial
para conclusão do Curso de Engenharia de Computação

Prof. Dr. Thiago de Paulo Faleiros (Orientador)
CIC/UnB

Prof. Vinicius Ruela P. Borges
CIC/UnB

Prof. Luís Paulo Faina Garcia
CIC/UnB

Prof. Dr. José Edil Guimarães de Medeiros
Coordenador do Curso de Engenharia de Computação

Brasília, 24 de junho de 2019

Dedicatória

Dedico este trabalho à memória de Patrick Joseph Falterman II, meu amigo, que me ensinou que o maior perigo que uma pessoa pode correr é não sair do lugar.

"*Imagine the moment you realize that waiting has made you old and unfit to seek the adventure you always dreamed of. now... Imagine the Amazon.*"
— **Things in Squares**, The Modern Nomad

Agradecimentos

Agradeço ao Estado Brasileiro que, através de políticas de inclusão, tornou possível o ingresso deste aluno de Escola Pública ao tão sonhado curso de Engenharia de Computação. Agradeço, também, à comunidade de Software Livre, por fornecer gratuitamente as ferramentas necessárias para que eu desenvolvesse este trabalho.

Resumo

A avaliação de modelos não supervisionados é um processo crítico na descoberta de conhecimento. É intríseco ao contexto não supervisionado o não conhecimento de rótulos pré-estabelecidos e, em muitos casos, os padrões são determinados pelo próprio processo algorítmico. Quando se trata de documentos no formato textual, uma técnica bastante utilizada no aprendizado não supervisionado são os *Modelos de tópicos*. Modelos de Tópicos são um conjunto de algoritmos que tem como função inferir, a partir de uma grande quantidade de documentos textuais, os temas neles contidos. Este arcabouço de técnicas é bastante utilizado para a sumarização, exploração e classificação de documentos. Em geral, o produto destes modelos é um conjunto de tópicos (temas) e sua distribuição sobre os documentos processados. Cada tópico é constituído por um conjunto de palavras com diferentes probabilidades de ocorrência. Devido à característica não-supervisionada destes algoritmos, nem sempre os tópicos gerados são formados por palavras semânticamente relacionadas, ou seja: tópicos aprendidos podem não fazer sentido para um leitor humano. Estes tópicos têm pouca utilidade na análise de documentos por não possuírem a interpretabilidade necessária para representar um assunto real. Para identificar estes tópicos pouco úteis, a avaliação manual por especialistas humanos pode ser utilizada, porém, esta é uma atividade onerosa e lenta. Por isso, algumas técnicas de avaliação automática foram propostas. As técnicas automáticas mais bem estabelecidas na literatura consistem em buscar por co-ocorrências de pares de palavras dentro de uma grande base de conhecimento, comumente a Wikipedia. Devido ao tamanho da base de busca, existem problemas de lentidão e excessivo gasto computacional. Assim, neste trabalho foi investigada a aplicabilidade de palavras imersas no espaço vetorial para avaliar tópicos de forma mais ágil e eficiente. Os resultados obtidos basearam-se na correlação entre as técnicas que utilizam vetores associados à palavras com as técnicas de avaliação automática baseadas na co-ocorrência entre pares de palavras.

Palavras-chave: Modelos Probabilísticos de Tópicos, Word Embeddings, Pointwise Mutual Information, Word2Vec

Abstract

Evaluating unsupervised models is a critical process on knowledge discovering. Nonexistent label assignment is intrinsic to unsupervised context. In many cases, patterns are determined by the algorithmic process itself. Regarding to textual documents, a technique largely used for unsupervised learning are *Topic Models*. Topic Models are a group of algorithms used to estimate from a large number of textual documents their thematic structure. This framework of techniques is widely used for document automatic categorization and exploration, with many possible applications. In general, these models result in a set of topics (themes) and their distribution over the training documents. Each topic consists of a set of words with different possibility of occurrence. Due to the unsupervised characteristic of these algorithms, the generated topics are not always formed by semantically related words. Some of the topics can make no sense to a human reader, therefore, these poorly interrelated topics are less useful for document analysis as they do not group documents semantically. Human evaluation can be used to identify these weak topics, but this is a very expensive and slow task. To solve this problem, some automatic evaluation techniques were proposed in the literature. The techniques that obtained best results consist of searching for co-occurrence of words-pairs inside external data sources, commonly the Wikipedia. Due to the size of the data source, problems of slowness and expensive computational cost are found. Thus, in this work the application of word embedding on topic evaluation was investigated for better performance and efficiency. The obtained results were compared with the prior techniques by means of correlation analysis.

Keywords: Topic Models, Word Embedding, Pointwise Mutual Information, Word2Vec, Topic Coherence

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

LDA Latent Dirichlet Allocation.

NILC Núcleo Interinstitucional de Linguística Computacional-NILC/USP.

NPMI Normalized Pointwise Mutual Information.

PMI Pointwise Mutual Information.

Capítulo 1

Introdução

Grande parte do conteúdo que trafega hoje pela internet está no formato textual. *Tweets*, postagens do *Facebook* e comentários dos mais variados são compartilhados por bilhões de usuários ao redor do mundo diariamente. Além disso, mídias que anteriormente eram produzidas e distribuídas fisicamente (notícias de jornal, artigos científicos, livros etc.) hoje estão migrando para o mundo digital. Produtores distribuem seus conteúdos através de portais, leitores digitais e periódicos on-line. De olho nesta mudança, empresas ao redor do mundo também têm investido milhões na conversão de documentos físicos para o formato digital. Portanto, a quantidade de dados armazenados e produzidos no formato de linguagem natural cresce exponencialmente na medida em que a quantidade de usuários dos meios digitais também cresce.

Um dos maiores desafios nesta área de Processamento de Linguagem Natural é a sumarização e exploração de grandes quantidades de documentos [6]. Conforme a quantidade de dados aumenta, a avaliação e análise individualizada de documentos por humanos se torna uma tarefa cada vez mais onerosa e impossível de ser realizada sem ajuda da tecnologia. Por isso, técnicas de agrupamento automático, busca e exploração inteligente de documentos são ferramentas muito importantes nesta área.

Os *Modelos Probabilísticos de Tópicos* foram desenvolvidos para auxiliar nesta tarefa [6]. Trata-se de um grupo de algoritmos que buscam inferir, a partir das palavras contidas nos documentos, a estrutura temática dos documentos. Seu objetivo é estimar quais são os temas abordados por aquele conjunto de documentos e como estes temas se interconectam.

Os tópicos gerados representam os assuntos/temas abordados por cada conjunto de documentos. Cada tópico é constituído por uma distribuição de probabilidade sobre todas as palavras. Observando-se as palavras com maior probabilidade de ocorrência em cada tópico é esperado que se consiga inferir qual é o assunto relacionado àquele tópico. Um tópico é um conjunto de palavras que ocorrem frequentemente em documentos semanti-

camente relacionados e que podem ser usadas para descrever o assunto ou tema que os documentos tratam. Aqui, de forma prática, um tópico k é o conjunto de palavras formada pelas $top - X$ palavras melhores ranqueadas na distribuição de tópicos por palavras.

Porém, devido à característica totalmente automática e não supervisionada destes algoritmos, modelos probabilísticos de tópicos podem aprender tópicos formados por palavras fracamente relacionadas semânticamente, ou seja, palavras que não agrupam semânticamente documentos. Estudos já demonstraram que identificar estes tópicos pouco úteis pode melhorar a performance destes modelos [7].

Várias métricas de avaliação de modelos de tópicos existiam na literatura [8], porém, foi demonstrado que a avaliação semântica das palavras que constituem os tópicos gerados tem maior correlação com a análise por especialistas do que as métricas que somente analisavam os modelos internamente [9]. As métricas que avaliam internamente são cálculos de uma função objetiva que, no caso dos modelos probabilísticos, estão relacionadas com o valor da distribuição conjunta do modelo. Uma métrica de avaliação de modelos proposta é baseada no cálculo Pointwise Mutual Information (PMI), avaliando a co-ocorrência das palavras dos tópicos em uma fonte de conhecimento, e obteve ótimos resultados [10]. Um grande estudo de comparação entre diversas técnicas de avaliação de tópicos concluiu que esta técnica, utilizando a Wikipedia como fonte de conhecimento, apresentava o melhor desempenho na correlação com a avaliação por humanos [11].

Além disso, recentemente foi proposta uma nova técnica de vetorização de palavras através das técnicas skip-gram e continuous bag-of-words, também conhecidos como Word2Vec [3]. Uma fonte suficientemente grande de conhecimento, como a Wikipedia, é utilizada para treinar uma rede de vetores, cada um representando um termo do vocabulário, resultando na representação vetorial das palavras contidas naquele vocabulário. Devido à forma com que a rede ajusta estes vetores (aproximando palavras que aparecem no mesmo contexto), é esperado que as distâncias entre estes vetores estejam em conformidade com as relações semânticas entre as palavras, permitindo diversas operações algébricas. Anteriormente, as técnicas de vetorização eram muito mais custosas computacionalmente, feitas por redes neurais profundas [2], e não apresentavam resultados tão bons quanto os observados por esta nova técnica. Isto faz do Word2Vec uma ferramenta poderosíssima que ainda pode ter diversas utilidades em estudos futuros em processamento de linguagem natural.

Com isso, tem-se a seguinte hipótese de pesquisa: a correta combinação de vetores de palavras imersos no espaço euclidiano podem ser utilizados como uma função de ranqueamento de tópicos.

1.1 Objetivos

Investigar a aplicação de vetores de palavras na avaliação semântica de tópicos por meio da correlação com as técnicas tradicionais de avaliação e, com isso, obter melhor desempenho. Para tal, modelos e tópicos individualmente serão avaliados utilizando a técnica proposta e os resultados serão comparados com os métodos anteriores através do cálculo estatístico de correlação, e os resultados serão analisados. A hipótese a ser testada é, então, de que é possível avaliar tópicos utilizando vetores de palavras. Além disso, como efeito colateral deste trabalho, será implementada na língua portuguesa uma ferramenta de avaliação automática de modelos de tópicos. A ferramenta desenvolvida consistirá em uma API que avalia modelos pelas métricas PMI e sua versão normalizada. [10] [1] utilizando como fonte de conhecimento a Wikipedia em português.

1.1.1 Estrutura do trabalho

No Capítulo 2 deste trabalho encontra-se a fundamentação teórica, com os principais conceitos necessários para o entendimento do projeto. No Capítulo 3, a metodologia e as ferramentas utilizadas e desenvolvidas para realização dos experimentos. No Capítulo 4 os resultados são expostos e discutidos. Por fim, no Capítulo 5 encontra-se a conclusão e encaminhamentos futuros deste trabalho.

Capítulo 2

Revisão Bibliográfica

2.1 Modelos de Tópicos

Modelos de Tópicos são um conjunto de algoritmos não supervisionados que, a partir das palavras contidas em uma grande quantidade de documentos, extraem os temas abordados por eles. Um dos seus objetivos é conseguir agrupar, relacionar e analisar documentos a partir da observação da sua distribuição sobre os tópicos/temas preservando as relações estatísticas e semânticas entre eles [12] [6].

Cada documento é modelado como um vetor numérico de V dimensões, sendo V a quantidade de palavras do vocabulário. Cada posição do vetor está relacionada à presença de uma palavra no documento, seu conteúdo pode representar a quantidade de vezes que uma palavra aparece, a informação binária de presença ou não da palavra ou qualquer outra forma de representar a significância daquela palavra para o documento. Desta forma, um corpus qualquer pode ser modelado como uma matriz $M \times V$ (M = quantidade de documentos).

O resultado da aplicação de um Modelo de Tópicos sobre um conjunto de documentos é, então, a representação destes documentos em um espaço dimensional muito menor $M \times K$ (K = quantidade de tópicos). Este espaço reduzido pode ser chamado também de *Espaço Latente* [9], pois é esperado que este preserve as relações ocultas (latentes) entre os documentos. Uma matriz $K \times V$ de distribuição de tópicos sobre palavras também é obtida, e a análise das palavras que constituem cada tópico é importante para interpretar os temas reais identificados.

Os tópicos são usualmente representados pelas palavras mais significantes em sua distribuição. Para exemplificar, aplicamos o modelo Latent Dirichlet Allocation (LDA) [12] para encontrar tópicos em um dataset de 100 mil notícias em português. Representando os tópicos pelas 5 palavras mais significativas, estes foram alguns dos resultados encontrados:

$$\{universidade, professor, usp, pesquisa, instituto\}$$

$$\{banda, show, rock, grupo, disco\}$$

$$\{avião, aeroporto, viagem, vôo, viagens\}$$

Note que é fácil identificar os temas reais com os quais estes tópicos se relacionam. Este tipo de análise demonstra como os Modelos de Tópicos podem ser úteis e poderosos na exploração, descrição e análise de documentos textuais.

2.1.1 Latent Dirichlet Allocation

Em 2003, *David M. Blei* propôs o modelo probabilístico Latent Dirchlet Allocation [12], algoritmo de extração de tópicos, que viria a se tornar o modelo mais difundido e base para diversas modificações [6]. O modelo se utiliza do **processo generativo** para descrever a forma como documentos poderiam ser gerados a partir de distribuições conhecidas. Estas distribruições são:

Distribuição de Tópicos: Cada um dos tópicos é uma distribuição discreta de *Dirichlet* sobre todas as palavras do vocabulário. Esta distribuição tem um fator λ de concentração, que garante que um pequeno subconjunto de palavras terá maior probabilidade de ocorrência em cada tópico. A quantidade **K** de tópicos é definida pelo usuário.

Distribuição de Documentos: Cada documento é uma distribuição discreta de *Dirichlet* sobre todos tópicos, que pretende refletir o assunto tratado por cada texto. Novamente a distribuição de Dirichlet é utilizada para garantir que um pequeno subconjunto de tópicos terá maior probabilidade para cada documento.

O **processo generativo**, que explica conceitualmente como cada documento é constituído, se inicia por selecionar aleatoriamente um tópico dentro da distribuição sobre tópicos do documento (**Distribuição de Documentos**). Em seguida, seleciona-se uma palavra aleatoriamente dentro do tópico encontrado no passo anterior (**Distribuição de Tópicos**). Por fim, a palavra é inserida no documento e o processo se repete até que o documento esteja completo.

Este processo pode ser descrito mais formalmente definindo-se $\beta_{1:K}$ como a distribuição dos K tópicos sobre palavras, sendo β_k a distribuição do tópico k. $\theta_{1:D}$ é a distribuição dos D documentos sobre tópicos, onde $\theta_{d,k}$ é a probabilidade do documento d pertencer ao tópico k sobre todas as palavras. $z_{1:D}$ é a atribuição de tópicos por documento, $z_{d,n}$ indicando o tópico que gerou a palavra n no documento d. Já as palavras observadas são

representadas por $w_{1:D}$, com $w_{d,n}$ a palavra de número n no documento d. O processo generativo pode ser representado pela seguinte equação [6]:

$$p(\beta_{1:K}, \theta_{1:D}, z_{1:D}, w_{1:D}) = \prod_{i=1}^{K} p(\beta_i) \prod_{d=1}^{D} p(\theta_d) \left(\prod_{n=1}^{N} p(z_{d,n}|\theta_d) p(w_{d,n}|\beta_{1:K}, z_{d,n}) \right)$$

O processo generativo é uma premissa interessante para se entender o funcionamento e a utilidade das distribuições do LDA. Entretanto, no mundo real, a única variável conhecida são os documentos e suas palavras. As distribuições de documentos sobre tópicos e de tópicos sobre palavras são as variáveis que queremos descobrir. Ajustando a equação para o caso real, em que apenas $w_{1:D}$ é conhecido, temos:

$$p(\beta_{1:K}, \theta_{1:D}, z_{1:D}|w_{1:D}) = \frac{p(\beta_{1:K}, \theta_{1:D}, z_{1:D}, w_{1:D})}{p(w_{1:D})}$$

Calcular diretamente esta probabilidade é intratável computacionalmente devido às múltiplas combinações possíveis de distribuições [6]. Este problema é resolvido aplicando-se algumas técnicas de aproximação como, por exemplo, o Amostrador de Gibbs [6]. Algoritmo que inicia as distribuições com números aleatórios e os aproxima iterativamente através de amostras dos documentos. Outro método comum é o de Inferência Variacional [12].

2.2 Avaliação de Modelos de Tópicos

2.2.1 Perplexidade e likelihood

Os primeiros métodos de avaliação dos modelos probabilísticos de tópicos eram baseados no calculo da *perplexidade* [12]. Os modelos eram avaliados a partir da probabilidade atribuída a documentos de teste. A função de perplexidade é equivalente algébrico do inverso da média da probabilidade (*log-likelihood*) por palavra. Considerando M documentos, em que D_{teste} são os documentos de teste, $\mathbf{w}_d$ o vetor de palavras contidas no documento d e N_d o número de palavras no documento d, a funcão de perplexidade obedece a seguinte equação: [12]

$$perplexidade(D_{teste}) = exp\left\{ -\frac{\sum_{d=1}^{M} logp(\mathbf{w}_d)}{\sum_{d=1}^{M} N_d} \right\}$$

Quanto maior a probabilidade atribuída a documentos de teste, e portanto o gràu de confiança do modelo aumenta para documentos não vistos, mais bem avaliado era o modelo. Diversas técnicas de se estimar esta probabilidade de documentos não vistos

$\mathbf{w}_d$ foram discutidas no trabalho *Evaluation Methods for Topic Models* [8]. As técnicas descritas utilizavam ferramentas estatísticas para aprender como modelos poderiam ser aplicados a documentos não vistos. Porém, a representação interna dos modelos era ignorada [9].

2.2.2 Avaliação Semântica de Tópicos

Em *Reading Tea Leaves: How Humans Interpret Topic Models* [9], *Chang J.* provocou uma grande mudança na área de avaliação de modelos de tópicos ao propor novas formas de avaliação a partir da **coesão semântica dos tópicos**. Anteriormente, as palavras que constituem os tópicos eram interpretadas e utilizadas para identificar os temas representados. Implicitamente, era esperado que estas palavras possuíssem algum significado semântico, porém, neste trabalho, pela primeira vez métodos quantitativos de avaliar a coesão de tópicos foram propostos. Foram eles:

Intrusão de Palavras: Palavras são aleatoriamente adicionadas a tópicos e, então, é solicitado para que avaliadores humanos identifiquem tais palavras "intrusas". O grau de facilidade com que tais palavras são identificadas é utilizado para avaliar a coesão semântica dos tópicos.

Intrusão de Tópicos: O resumo de um documento é apresentado para um avaliador humano. Depois, um conjunto de tópicos é apresentado a ele, dentre os quais um é aleatoriamente selecionado dentre os menos significativos para o documento. A facilidade com que o avaliador identifica tal tópico é utilizada para avaliar a coesão semântica do modelo.

Modelos eram avaliados a partir da performance que seus tópicos e documentos atribuídos apresentavam nestas técnicas.

Foi verificado que modelos bem avaliados por coesão de tópicos tendiam a ser mal avaliados pelo cálculo de perplexidade. Identificando, assim, uma **correlação negativa em relação às técnicas de avaliação anteriores**. Por isso, as técnicas que se seguiram na literatura buscavam automatizar esta análise semântica de tópicos.

2.2.3 Pointwise Mutual Information (PMI)

No trabalho *External Evaluation of Topic Models* [10], algumas formas de avaliação automática de tópicos foram propostas. Com o objetivo de se avaliar a coesão semântica das palavras formadoras dos tópicos, técnicas foram propostas e comparadas com a avaliação por humanos. Todas elas combinavam as palavras mais importantes dos tópicos em pares, e calculavam sua co-ocorrência em bases de conhecimento externas.

A técnica que obteve os melhores resultados baseia-se no cálculo de Pointwise Mutual Information (PMI) de todos os pares de palavras utilizando a Wikipedia como fonte de conhecimento.

O cálculo PMI é feito da seguinte maneira: a base de conhecimento é considerada como um documento único. Então, as probabilidades de ocorrência de cada par de palavras são calculadas considerando-se uma janela deslizante de n-palavras sobre a base externa (o autor utiliza uma janela de 10 palavras no trabalho). A Figura 2.1 demonstra de que forma a janela deslizante funciona, com pares de palavras sendo buscados cada vez que a janela desliza, neste caso, procurando as palavras 'buraco' e 'negro'.

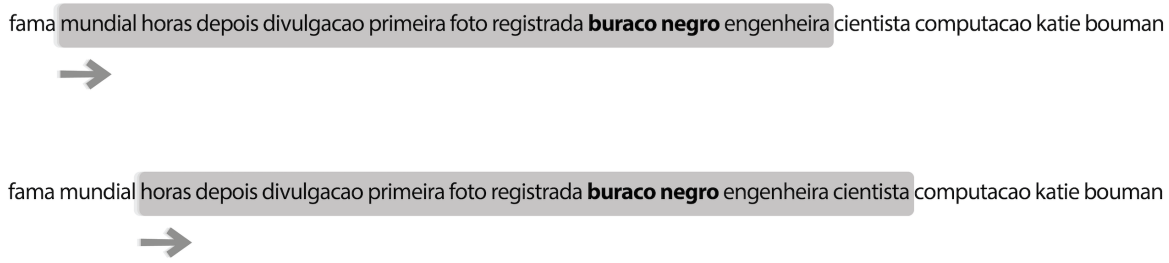

Figura 2.1: Janela deslizante utilizada no cálculo PMI.

Portanto, para um tópico $\mathbf{w}$, representado por uma lista de n palavras mais prováveis na distribuição de um tópico k:

$$topico = \mathbf{w} = [w_1, w_2, w_3, w_4, w_5, w_6, w_7, w_8, w_9, w_{10}]$$

$$PMI\text{-}topico(\mathbf{w}) = mediana\{PMI(w_i, w_j) : w_i, w_j \in \mathbf{w}\}$$

$$PMI(w_i, w_j) = \log \frac{P(w_i, w_j)}{P(w_i)P(w_j)}$$

No trabalho, o autor avalia tópicos automaticamente utilizando a Wikipedia em inglês e a base Google-n-grams. Os tópicos foram gerados a partir de uma base de notícias em inglês, e outra de livros em inglês. Os resultados foram comparados com análises realizadas por 9 avaliadores humanos.

O cálculo PMI utilizando a Wikipedia como base de conhecimento foi o que obteve correlações mais consistentes com avaliadores humanos, acima de 0.7. Considerada alta, dado que a intercorrelação entre os avaliadores foi de 0.78-0.81. Também foi demonstrado que a tarefa de comparação de similaridade entre documentos melhora sua performance ao se considerar apenas os tópicos melhor avaliados por PMI. Em comparação com a

avaliação por humanos, a correlação obtida nesta tarefa chegou a 0.94 utilizando-se apenas os melhores tópicos.

2.2.4 Outras técnicas utilizando Google, Wordnet e Wikipedia

Várias outras técnicas de avaliação de modelos e tópicos foram propostas e comparadas entre si na literatura. Aqui, apresentaremos um trabalho que comparou várias delas com avaliações feitas por humanos [11]. Nele, algumas técnicas utilizando Google, Wordnet e Wikipedia foram comparadas.

Wordnet

Wordnet é um banco de dados aberto que armazena informações de diversos termos léxicos do inglês. Os termos são agrupados em conjuntos de *synsets* (ou sinônimos). Estes *synset* são interligados de acordo com relações semânticas e conceituais. Resultando numa estrutura da qual é possível se extrair relações semânticas entre palavras [13].

Para avaliação de tópicos, o estudo propôs diversas técnicas de comparação de palavras em pares, seja pela distância entre *synsets*, número de pulos necessários entre *synsets*, etc. Para isso, as relações hiperonímicas entre termos foram utilizadas, ou seja, agrupamentos hierárquicos de palavras que indicam se estas pertencem a um mesmo conceito.

Wikipedia

Wikipédia é uma enciclopédia livre, disponível na internet em diversas línguas e completamente aberta e editável por qualquer colaborador. Atualmente, sua versão em português possui mais de 1 milhão de artigos e mais de 5 mil usuários ativos [14].

No artigo em discussão, a versão em inglês foi utilizada para fazer comparação de termos par a par. Algumas técnicas buscaram verbetes/artigos cujo título fosse a própria palavra, outros utilizaram a Wikipédia como fonte de conhecimento textual e procuraram os pares de palavras no corpo de todos os artigos.

- Milne-Witten (MIW) - Conta os hiperlinks entre os dois artigos, considerando a quantidade total de hiperlinks para cada verbete.
- Related Article Concept Overlap (RACO) - Analisa os hiperlinks externos dos dois artigos, classificando-os por categoria e analisa a intersecção de categorias dos dois verbetes.
- Document Similarity (DOCSIM) - Simples comparação textual de distância entre os dois artigos.

- Pointwise Mutual Information - A Wikipedia inteira é considerada como um único documento e calcula-se a probabilidade de co-ocorrência dos pares dentro de uma janela de 10 palavras através do PMI.

Google

Através do buscador, as palavras dos tópicos eram agrupadas como uma *query* única para busca em toda a internet indexada pelo Google. Portanto, esta técnica não avaliava as palavras de par a par, mas sim todas as palavras juntas. Duas métricas foram utilizadas:

- LOGHITS - Logaritmo do número total de resultados retornados pelo buscador.
- TITLES - Ocorrência das palavras nos títulos dos 100 primeiros sites retornados.

Resultados de Comparações

Para comparar todas as técnicas apresentadas, dois conjuntos de tópicos foram utilizados. Um foi extraído de um *dataset* de livros, outro de um *dataset* de notícias. Foi pedido que humanos avaliassem os tópicos com nota de 1 a 3 por sua "utilidade". Então, a correlação entre as técnicas automáticas e os avaliadores humanos foi computada.

O cálculo de Pointwise Mutual Information (PMI) utilizando a Wikipedia foi o que obteve os melhores e mais consistentes resultados, apresentando correlações muito próximas da intercorrelação entre avaliadores humanos (IAA) [1]. Na Figura 2.2 podem ser observados os resultados de tais experimentos.

Resource	Method	Median	Mean
WordNet	HSO	−0.29	0.34
	JCN	0.08	0.22
	LCH	−0.18	−0.07
	LESK	0.38	0.37
	LIN	0.18	0.25
	PATH	0.19	0.11
	RES	−0.10	0.13
	VECTOR	0.07	0.20
	WUP	0.03	0.10
Wikipedia	RACO	0.61	0.63
	MIW	0.69	0.60
	DOCSIM	0.45	0.50
	PMI	0.78	0.77
Google	TITLES	**0.80**	
	LOGHITS	0.46	
Gold-standard	IAA	0.79	0.73

Resource	Method	Median	Mean
WordNet	HSO	0.15	0.59
	JCN	−0.20	0.19
	LCH	−0.31	−0.15
	LESK	0.53	0.53
	LIN	0.09	0.28
	PATH	0.29	0.12
	RES	0.57	0.66
	VECTOR	−0.08	0.27
	WUP	0.41	0.26
Wikipedia	RACO	0.62	0.69
	MIW	0.68	0.70
	DOCSIM	0.59	0.60
	PMI	**0.74**	**0.77**
Google	TITLES	0.51	
	LOGHITS	−0.19	
Gold-standard	IAA	0.82	0.78

Figura 2.2: Correlação com avaliadores humanos para tópicos de notícias (esquerda) e tópicos de livros (direita) [1].

2.2.5 Normalized Pointwise Mutual Information (NPMI)

Em *Machine Reading Tea Leaves: Automatically Evaluating Topic Coherence and Topic Model Quality* [1], foi realizado mais um trabalho extenso de comparação entre técnicas de avaliação automática de tópicos. Desta vez, foi investigada a automatização da identificação de palavras intrusas em tópicos. Comparações e avaliações foram feitas em vários níveis, tanto avaliando tópicos diretamente, quanto modelos pelos *scores* de seus tópicos, mas sempre comparando com avaliações feitas por humanos.

No artigo, foi incluída uma versão normalizada do PMI [1], baseando-se no artigo *Normalized Pointwise Mutual Information* [15]. Esta normalização é feita para se retirar o viés do PMI, que favorece palavras com baixa frequência no corpus. Na equação NPMI, a probabilidade de co-ocorrência das palavras w_i e w_j é dada pelo PMI dividido por $-\log P(w_i, w_j)$:

$$NPMI(w_i, w_j) = \frac{PMI(w_i, w_j)}{-\log P(w_i, w_j)}$$

Nos resultados deste trabalho, a avaliação a nível de modelo obteve maior correlação com análise por humanos quando feita através de NPMI utilizando a Wikipedia como base de conhecimento.

2.3 Representação de palavras

Um dos grandes desafios para se lidar computacionalmente com linguagem natural é a representação de palavras. Vocabulários podem chegar à casa de milhões de palavras e, além disso, documentos reais podem ter um número enorme de possíveis combinações entre elas. Imagine, por exemplo, que se queira modelar a probabilidade de ocorrência de uma sequência de 10 palavras em um vocabulário V de tamanho $100,000$. Considerando as palavras como variáveis independentes, a probabilidade seria de $1/100.000^{10} = 1/10^{50}$. Porém, sabe-se que as palavras não tem igual probabilidade de ocorrência e, também, grupos de palavras tem maior probabilidade de ocorrer conjuntamente. Este é o atual desafio para representação de palavras em vetores numéricos entendíveis pelo computador [2].

Diversas formas de representação de palavras já foram utilizadas. Algumas tão somente para especificar quais palavras estão presentes em determinado documento, outras mais recentes pretendem representar relações semânticas mais complexas entre palavras.

2.3.1 One-hot Encoding

Em processamento de linguagem natural, *one-hot encoding* é a forma mais direta de se representar palavras. Aqui, cada palavra é codificada como um vetor que possui o número 1 em apenas um de seus índices, e o número 0 em todos os outros. Desta forma, palavras são convertidas em vetores unitários de dimensão $|V|$, em que cada dimensão do vetor tem relação com apenas uma palavra do vocabulário [16]. Exemplo:

$frase =$ 'Cientistas revelam imagem de buraco negro.'
$vocabulario = (imagem, buraco, negro, cientistas, revelam)$
$imagem = [1, 0, 0, 0, 0]$
$buraco = [0, 1, 0, 0, 0]$
$negro = [0, 0, 1, 0, 0]$
$cientistas = [0, 0, 0, 1, 0]$
$revelam = [0, 0, 0, 0, 1]$

Problemas com a codificação One-hot encoding

- **Ausência de valor semântico** - Vetores são ortogonais entre si, dificultando operações algébricas. Por definição, a disposição dos vetores não reflete nenhuma relação entre as palavras.
- **Tamanho** - Vetores podem ficar muito grandes, com vocabulários chegando facilmente à casa dos milhões de palavras.

Apesar destes problemas, *one-hot encoding* é uma ferramenta útil para se converter documentos textuais em representações numéricas, facilitando seu processamento.

2.3.2 Saco de Palavras

Bag-of-Words, ou saco-de-palavras, é a representação de documentos através do conjunto (saco) das palavras neles contidos, sem levar em consideração a ordem ou relações entre as palavras, apenas a quantidade de vezes em que elas aparecem. Pode ser considerado, também, como a somatória do *one-hot encoding* de todas as palavras do documento [16]. Utilizando o exemplo anterior:

$frase =$ 'Cientistas revelam imagem de buraco negro.'
$vocabulario = (imagem, buraco, negro, cientistas, revelam)$
"imagem imagem" $= [2, 0, 0, 0, 0]$
"imagem buraco negro" $= [1, 1, 1, 0, 0]$
"cientistas revelam imagem" $= [1, 0, 0, 1, 1]$

Esta é uma ferramenta muito utilizada para converter documentos em vetores numéricos, que podem ser usados como features para treinar modelos de aprendizado de máquina, como por exemplo acontece com o LDA.

2.3.3 Tf-Idf

Term frequency-inverse document frequency, ou tf-idf, é uma medida estatística para calcular a importância que determinada palavra tem para um documento dentro de um corpus. Em oposição à representação *bag-of-words*, que apenas computa a frequência das palavras nos documentos, tf-idf diminui o peso de palavras muito comuns em todos os documentos do corpus. Assumindo w como o termo, d o documento a ser encodado e M como o conjunto de todos os documentos do corpus; tf-idf pode ser descrito de forma generalizada como:

$$tfidf(w, d, M) = tf(w, d) * idf(w, M)$$

onde $tf(w, d)$ é uma função que calcula a frequência do termo no documento e $idf(w, M)$ uma funcão que retorna o inverso da frequência do termo no corpus [17]. Por exemplo:

$$tf(w, d) = |\{i \in d : i = w\}|$$
$$idf(w, M) = \log \frac{|M|}{|\{d \in M : w \in d\}|}$$

onde $|M|$ é o número total de documentos e $|\{d \in M : w \in d\}|$ é a quantidade de documentos em que o termo w aparece. Note que tf-idf adiciona um multiplicador ao cálculo puro da frequência. Diminuindo o peso das palavras que, por serem muito comuns, provavelmente não adicionam considerável valor semântico ao documento.

2.3.4 Modelos de Semântica Distribucional

Em seu trabalho *Distributional Structure* [18], Zellig S. Harris descreve a hipótese distribucional. Segundo ela, o sentido das palavras pode ser obtido através do seu contexto, ou seja: uma palavra pode ter seu significado definido por aquelas com as quais ela frequentemente co-ocorre. Isto deriva da concepção de que palavras não ocorrem arbritariamente em uma linguagem. De fato, cada palavra costuma aparecer a uma certa distância relativa de outras palavras específicas (ou grupos de palavras específicos) [18].

A partir desta concepção, vários trabalhos e técnicas foram desenvolvidos para representar palavras computacionalmente em um espaço contínuo, na tentativa de refletir tais aspectos das linguagens. De posse dessas representações, seria possível comparar palavras

da mesma forma que se compara vetores, obtendo informações importantes quanto à sua similaridade e demais relações semânticas.

É importante salientar que esta área de estudos envolve dois grandes grupos de algoritmos e abordagens que são frequentemente confundidos: as redes neurais utilizadas para estimar vetores de representação de palavras (*word embedding* como *Word2Vec*), de que trataremos logo mais neste trabalho, e os modelos de semântica distribucional, de que o presente tópico aborda.

Distributional semantic models (DSM's), ou modelos de semântica distribucional, são técnicas utilizadas para construir o espaço semântico no qual palavras podem ser representadas vetorialmente a partir dos contextos nos quais aparecem. Como explica Peter B. Tourney em *From Frequency to Meaning: Vector Space Models of Semantics* [19], ao se construir um DSM, primeiro se monta a matriz de frequências, calculando quantas vezes cada palavra aparece em cada contexto (aqui, contextos podem ser palavras, conjuntos de palavras, documentos, etc). Depois, se aplicam formas de balanceamento dos pesos encontrados, por exemplo tf-idf no caso de ser uma matriz termo-documento. Por fim, realiza-se a redução de dimensionalidade para facilitar a comparação dos vetores, um método comum é a decomposição em valores singulares [20].

Aplicação em avaliação de tópicos

O trabalho *Evaluating Topic Coherence Using Distributional Semantics* [21] aplicou modelos semânticos distribucionais na avaliação de tópicos e comparou com a avaliação por humanos. Nele, o espaço semântico foi construído utilizando a contagem da co-ocorrência de palavras em uma janela +5 e -5 palavras sobre a Wikipedia em inglês, ou seja, cada termo era considerado como co-ocorrendo com as 5 próximas palavras e as 5 palavras anteriores. Os cálculos de PMI e NPMI foram utilizados para balancear os vetores (note que isto é diferente da avaliação de tópicos diretamente por estes métodos). Portanto, cada dimensão refletia a co-ocorrência da palavra com outro termo do vocabulário. Também foi reduzida a dimensionalidade dos vetores para se considerar apenas as palavras presentes nos tópicos.

Os resultados demonstraram que utilizar o espaço semântico era ligeiramente melhor que a observação direta do PMI. Além disso, gerar os vetores balanceando-os através do cálculo NPMI foi um pouco melhor que por PMI.

Problemas com esta abordagem

- **Vetores muito grandes** - a matriz de frequências é formada por vetores usualmente muito grandes. Isto se deve ao tamanho do vocabulário e da quantidade de contextos possíveis.

- **Custo computacional** - Tanto para se calcular a matriz de frequências quanto para se fazer a redução de dimensionalidade, uma grande quantidade de computação é necessária.

2.4 Modelos de Linguagem em Redes Neurais

Bengio et. al. em *A Neural Probabilistic Language Model* propôs uma nova abordagem para o aprendizado (treinamento) de vetores de palavras a partir da arquitetura em redes neurais. Para combater o que chamou de "maldição da dimensionalidade" [2], a rede aprende vetores de dimensão pré-definida a partir de observações reiteradas de uma base de conhecimento.

Considerando que cada vetor consiste em um conjunto de K features que representam uma palavra, o autor considera uma matriz C de dimensão $V \times K$, como sendo a representação de todos os V vetores de palavras no vocabulário.

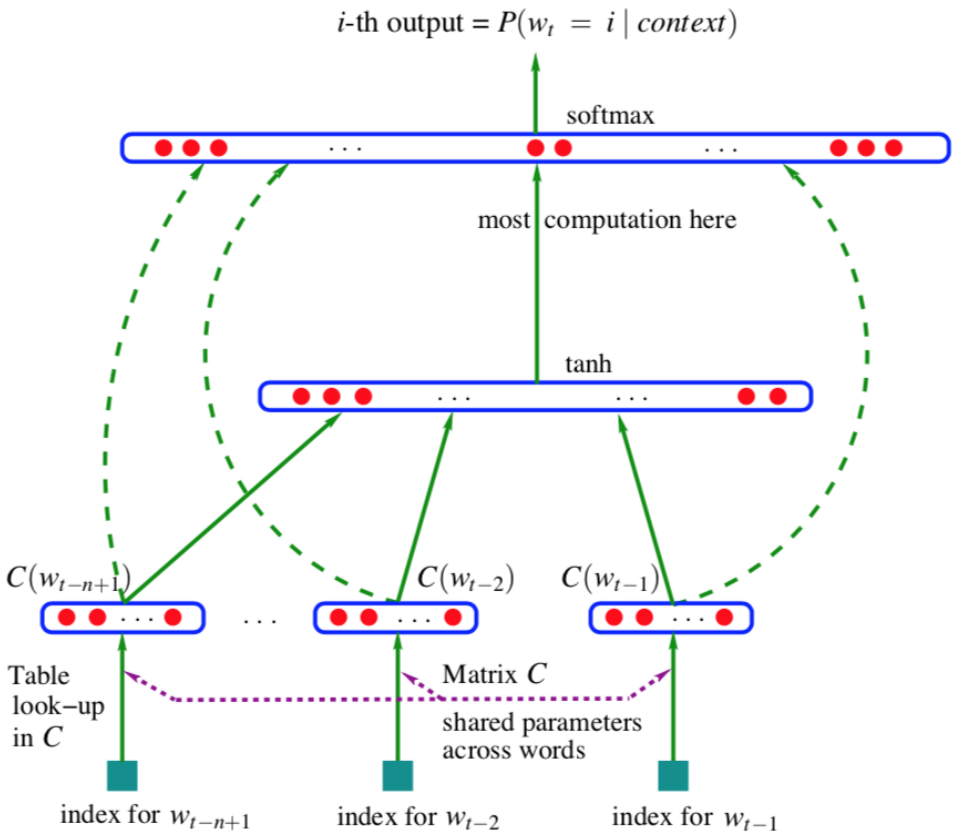

Figura 2.3: Arquitetura de rede neural proposta por Bengio et. al. [2]

Como se pode observar na Figura 2.3, na camada de entrada, uma sequência de n vetores em *one-hot encoding* são utilizadas para selecionar apenas as linhas da matriz C correspondentes às features das palavras do contexto, com t sendo a posição da base de conhecimento sendo observada para o treinamento.

Na camada oculta, uma função é aplicada recebendo a concatenação dos vetores de palavras x, esta função calcula um vetor y de dimensão $|V|$. Considerando b o *bias* da rede, W uma matriz opcional de ligação direta com a camada de saída (linha pontilhada na Figura 2.3), d o *bias* da camada oculta, H a matriz de pesos da camada oculta e U os pesos entre a camada oculta e a camada de saída. A camada foi modelada da seguinte maneira [2]:

$$y = b + Wx + Utanh(d + Hx)$$

Para fazer com que a saída seja um vetor de probabilidade, e cada índice i reflita a probabilidade condicional da i-ésima palavra dado o contexto, na saída é aplicada a função *softmax*. A função *softmax* normaliza a função (aqui utilizo uma função genérica y que recebe w) para refletir uma distribuição de probabilidades, garantindo que os resultados serão positivos e que sua soma seja 1 [2]:

$$P(w_t|w_{t-n+1}, ..., w_{t-1}) = \frac{e^{y_{w_t}}}{\sum_i e^{y_{w_i}}}$$

Com $\theta = (W, U, C, H, b, d)$ sendo os parâmetros e pesos da rede e ε o *learning rate*, a função de atualização de pesos a cada iteração se dá pelo *stochastic gradient ascent* [2], quando apresentada a palavra alvo w_t :

$$\theta \leftarrow \theta + \varepsilon \frac{\delta \log P(w_t|w_{t-n+1}, ..., w_{t-1})}{\delta \theta}$$

Note que a rede atualiza os pesos de forma a maximizar a probabilidade da palavra w_t dado o contexto em que esta aparece $(w_{t-1}, ..., w_{t-n+1})$ [2]. Isto é feito através da derivada do *log-likelihood* da função implementada. De posse desta derivada, os pesos são atualizados na direção que produz maior probabilidade para cada palavra w_t, obedecendo o *learning rate* ε que define o tamanho dos passos com que os pesos serão atualizados [22].

A aplicação da rede neural em testes de predição foram promissoras e demonstraram ter maior capacidade de generalização e aplicação em contextos maiores. Tendo em vista que os métodos anteriores possuíam séria restrição ao aumento do contexto (número de palavras utilizadas pra estimar vetores).

2.5 Word2Vec

Recentemente, foi proposto por *Tomas Mikólòv* o algoritmo que ficou popularmente conhecido como *Word2Vec* [3] [4], uma rede neural rasa para estimar representações contínuas de palavras. Seu diferencial é uma combinação de eficiência computacional muito superior em relação às anteriores e resultados práticos muito mais significativos em aplicações de similaridade e relações sintáticas/semânticas entre palavras.

Sob a premissa de que modelos mais simples treinados em quantidades muito grandes de dados poderiam obter resultados melhores que os modelos anteriores, mais complexos e por isso treinados em menores quantidades de dados, o autor propôs duas arquiteturas que eliminam a camada oculta não-linear:

2.5.1 Continuous Bag-of-Words

Utiliza como entrada as palavras de contexto que venham antes e depois da palavra alvo, a rede maximiza a probabilidade de predição correta da palavra do meio. Na camada de projeção, as palavras de contexto são somadas utilizando uma matriz de pesos $D \times V$ que consiste nos vetores de representação das V palavras em D dimensões. Por se tratar de uma soma simples dos vetores correspondentes às palavras contextos, a ordem em que as palavras aparecem não importa e, por isso, o nome *continuous bag-of-words*.

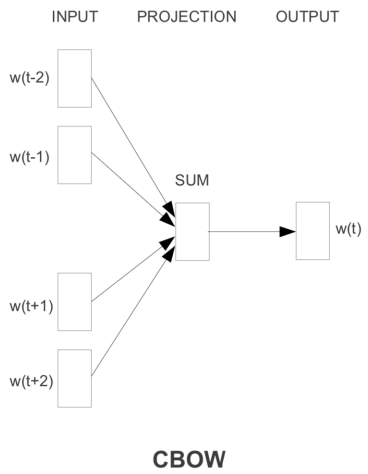

Figura 2.4: Arquitetura da rede neural *Continuous Bag-of-Words*. [3]

Para se calcular a camada de saída, e torná-la um vetor de probabilidades positivo, o autor utiliza uma modificação do *softmax*, chamada *hierachical softmax*. Para entender a

mudança, vamos dar uma olhada na função *softmax*, considerando v_c o vetor de contexto e v_w o vetor da palavra alvo:

$$P(w|c) = \frac{e^{v_c \cdot v_w}}{\sum_k^V e^{v_c \cdot v_k}}$$

Com V representando o número de palavras do vocabulário, que facilmente chega à casa dos milhões, pelo denominador da função *softmax*, nota-se que uma grande quantidade de computação é necessária já que N entradas são consultadas toda vez que uma probabilidade de saída é calculada.

A técnica *hierarchical softmax* converte o vocabulário em folhas de uma árvore binária que reflete a frequêcia das palavras no corpus (e.g. Árvore de *Huffman*), com palavras mais frequentes aparecendo mais próximas da raiz. O cálculo de *hierarchical softmax* faz a busca das palavra presentes no caminho entre o nó folha e a raiz, avaliando em média apenas $\log_2(N)$ nós.

2.5.2 Skip-Gram

Parecida com a arquitetura CBOW, a arquitetura *skip-gram* maximiza a probabilidade de predição das palavras do contexto a partir da palavra central [4].

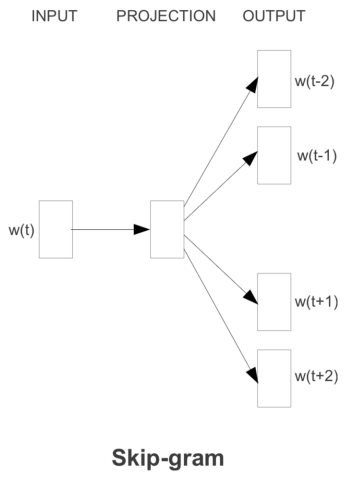

Figura 2.5: Arquitetura de rede neural *Skip-Gram* [4]

Com uma janela tamanho C, a rede seleciona aleatoriamente palavras entre $w - C$ e $w + C$ e maximiza a probabilidade dessas palavras dada a palavra central w, adicionando um fator de amortecimento pela distância.

2.5.3 Negative Sampling

Para otimizar ainda mais o funcionamento da rede, o autor propõe uma alternativa ao *hierarchical softmax*. O *negative sampling*, sempre que maximiza a probabilidade de palavras do contexto dada a palavra central, também minimiza a probabilidade de k palavras não presentes no contexto. Isto é feito para evitar que se consulte uma grande quantidade de palavras do vocabulário. Tomando σ como a função logística, o *negative sampling* funciona através da seguinte equação [4]:

$$\log P(w_c|w_I) = \log \sigma({v'_{w_c}}^T v_{w_I}) + \sum_{i=1}^{k} E_{w_i} \sim P_n(w)[\log \sigma(-{v'_{w_i}}^T v_{w_I})]$$

em que w_c é a palavra selecionada do contexto, w_I a palavra central de entrada. As k amostras negativas são aplicadas em $\log \sigma(-{v'_{w_i}}^T v_{w_I})$ e são retiradas de $P_n(w)$ que equivale à distribuição de probabilidade das palavras no corpus elevadas à potência 3/4, valor escolhido experimentalmente [4].

Note que duas matrizes de pesos são utilizadas, V e V', ambas de dimensão $D \times V$ e ambas representam as N palavras em vetores de dimensão D. As matrizes são somadas ao final do processamento.

Os vetores de palavras estimados através do treinamento de redes neurais *Word2Vec*, além de agrupar palavras que co-ocorrem nos mesmos contextos, também refletem relações semânticas e sintáticas entre palavras [4]. Esta característica foi importante para popularizar esta técnica.

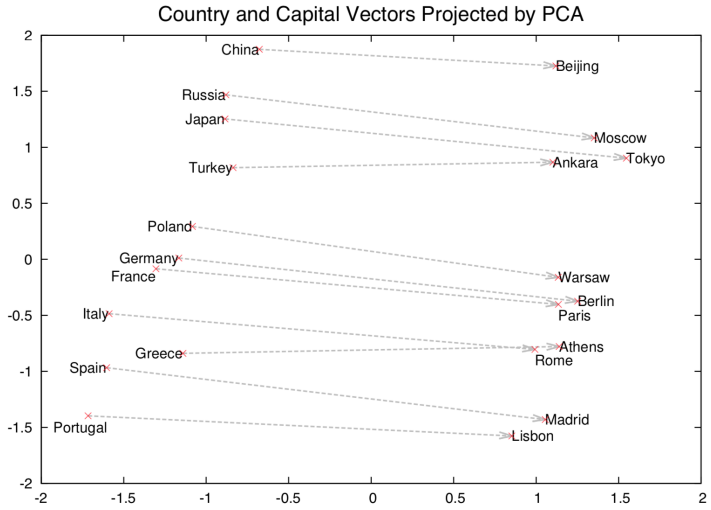

Figura 2.6: Vetores estimados por *Word2Vec* projetados em 2D [4].

Na Figura 2.6, por exemplo, os vetores de países e capitais têm distâncias similares entre si. De forma que $v(Poland) - v(Warsaw) \approx v(Germany) - v(Berlin)$. Ou seja, se subtrairmos o vetor 'Warsaw' do vetor 'Poland' e adicionarmos o vetor 'Berlin', os parâmetros que obteremos serão muito próximos do vetor 'Germany'.

2.6 Correlação Pearson

O coeficiente de correlação *pearson* é um método estatístico para medir a correlação linear entre duas variáveis [23]. De uso muito comum em todas as áreas da ciência, a medida de correlação Pearson r varia de -1 a $+1$, com 0 indicando nenhuma correlação entre as variáveis, $+1$ correlação linear positiva e -1 correlação linear negativa entre as variáveis. Portanto, este coeficiente serve para verificar o grau com que ambas variáveis tendem a ter o mesmo comportamento. A equação deste coeficiente se dá como segue:

$$r_{x,y} = \frac{cov(x,y)}{\sqrt{var(x)} \cdot \sqrt{var(y)}}$$

em que $cov(x,y)$ é a covariância entre as amostras das variáveis x e y, $var(x)$ a variância observada nas amostras de x e $var(y)$ a variância observada nas amostras de y [23]. Pode-se então extender a definição para:

$$r_{x,y} = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2} \sqrt{\sum_i (y_i - \bar{y})^2}}$$

onde $\bar{x}$ é a média das amostras da variável x e $\bar{y}$ a média das amostras da variável y.

Capítulo 3

Metodologia

O objetivo deste trabalho é investigar a aplicação de vetores de palavras estimados pela técnica *Word2Vec* [3], na avaliação de modelos de tópicos. Aqui, pretende-se obter melhores resultados do que aquelas técnicas encontradas na literatura (PMI [10] e NPMI [1]). Tais técnicas consistem em rotinas bastante custosas computacionalmente, percorrendo toda a base externa de conhecimento (normalmente a *Wikipedia*) em busca de co-ocorrências de palavras para cada avaliação. Já a utilização de *word embedding* implica que apenas uma rotina custosa computacionalmente será realizada: aquela em que se estimam os vetores de palavras. Portanto, toda vez que se avalia um tópico, apenas as relações entre os vetores já treinados serão consultadas, uma operação muito mais rápida. Desta forma, a utilização de *word embedding* traz um ganho considerável de performance nesta tarefa.

A forma mais recorrente de se avaliar a efetividade de novas ténicas de avaliação de tópicos é a correlação com a avaliação por humanos [10] [11] [1]. No referencial teórico deste trabalho alguns exemplos desta prática estão elencados. Neles se observa a utilização da correlação *Pearson* para comparar as notas dadas por avaliadores humanos com os valores de avaliação obtidos pelas técnicas automáticas. Entretanto, como a avaliação por humanos é uma tarefa custosa, e exigiria uma grande quantidade de pessoas para se obter uma amostragem estatística aceitável, optou-se por correlacionar as técnicas que utilizam *word embeddings* com as técnicas já consolidadas na literatura (PMI e NPMI) [23].

3.1 Tópicos gerados

Para a realização deste trabalho, o modelo probabilístico de tópicos *Latent Dirichlet Allocation* [12] foi aplicado quatro vezes sobre um corpus de notícias de jornal em português, gerando os tópicos que serão utilizados nas comparações. A escolha do LDA se deve aos ótimos resultados e várias aplicações encontradas nos referenciais literários da área

[6, 9, 24, 25]. Todos os tópicos foram gerados utilizando a biblioteca *gensim*[1] do *Python* que possui um empacotamento para uma outra implementação em Java e mais otimizada do LDA, *LDA-Mallet*[2].

3.1.1 Notícias (CHAVEFolha)

O corpus de notícias CHAVEFolha[3] é formado por 103.913 notícias publicadas pelo jornal 'A Folha de São Paulo' nos anos de 1994 e 1995. Neste trabalho, foi utilizado apenas o texto das notícias, rejeitando outras informações que não são necessárias à modelagem de tópicos.

No pré-processamento, foram retiradas as palavras de baixo valor semântico (stopwords) e todas as letras em caixa-alta foram convertidas para caixa-baixa (normalização). Além disso, foram removidos todos os símbolos e números, restando apenas as palavras contidas nas notícias.

A partir deste corpus, foram realizadas 4 rotinas de extração de tópicos utilizando o LDA. A quantidade de tópicos gerados foram de 100, 150, 200 e 250.

3.2 Vetores de palavras

Para realizar as comparações, foram estimados vetores de palavras através do modelo Word2Vec [3] utilizando a Wikipedia como base de conhecimento variando algumas configurações. Além disso, um segundo grupo de vetores de palavras pré-treinados em português também foi utilizado para as comparações, estes também foram estimados utilizando a técnica Word2Vec, porém, em um *dataset* bem maior.

Os vetores Word2Vec utilizados neste trabalho foram estimados utilizando a biblioteca *gensim* da linguagem de programação *Python*, através da classe *Word2Vec*. A escolha se deve à quantidade de parâmetros e configurações que podem ser alterados, atingindo assim comparações mais abrangentes.

Para estimar os vetores de palavras, foi utilizado o dataset da Wikipedia em português obtido da Wikimedia Dumps[4] formado por 932.010 verbetes desta enciclopédia virtual. Seu pré-processamento consistiu na conversão de todos os caracteres para caixa-baixa (normalização), remoção de palavras de baixo valor semântico (stopwords) e retirada de caracteres numéricos e símbolos, deixando apenas o conteúdo estritamente textual dos

[1] https://radimrehurek.com/gensim/
[2] http://http://mallet.cs.umass.edu/
[3] https://www.linguateca.pt/chave/
[4] https://dumps.wikimedia.org/ptwiki/

verbetes, seguindo assim o padrão adotado no trabalho *'Portuguese Word Embeddings'* [5].

Dois grupos de modelos foram gerados: aqueles que utilizam a técnica *Continuos Bag-of-Words* e aqueles que utilizam a técnica *Skip-gram*. Para ambos geramos vetores de 50, 100, 200, 300, 400, 500 e 1000 dimensões, com janela de contexto de 5 palavras. Além disso, fixando o modelo Skip-Gram de 300 dimensões, foram gerados vetores variando o tamanho da janela de contexto entre 4 e 10 palavras. Todos os modelos utilizaram a técnica de *Negative Sampling* para maior eficiência na estimação dos vetores. Palavras com menos de 5 ocorrências no corpus foram ignoradas neste processo.

Além dos vetores treinados apenas na Wikipedia, foram utilizados os vetores de palavras pré-treinados disponibilizados pelo Núcleo Interinstitucional de Linguística Computacional-NILC/USP (NILC)[5], esta base consiste num repositório de word embedding em português, estimados à partir de uma grande corpus de mais de 1.3 bilhões de palavras, envolvendo documentos de várias áreas do conhecimento. Tal repositório é resultado do trabalho *'Portuguese Word Embeddings: Evaluating on Word Analogies and Natural Language Tasks'* [5]. Todos os datasets utilizados para estimar estes vetores estão descritos na figura 3.1

Dentre todos os vetores disponibilizados pelo NILC, foram utilizados neste trabalho aqueles do tipo Word2Vec, estimados através das técnicas Skip-Gram e Continuous Bag-of-Words com 50, 100, 300, 600 e 1000 dimensões. Por terem sido estimados em corpus muito grandes e de domínios diversos, é esperado que se obtenha vetores de maior qualidade [5].

[5]http://www.nilc.icmc.usp.br/embeddings

Corpus	Tokens	Types	Genre	Description
LX-Corpus [Rodrigues et al. 2016]	714,286,638	2,605,393	Mixed genres	A huge collection of texts from 19 sources. Most of them are written in European Portuguese.
Wikipedia	219,293,003	1,758,191	Encyclopedic	Wikipedia dump of 10/20/16
GoogleNews	160,396,456	664,320	Informative	News crawled from GoogleNews service
SubIMDB-PT	129,975,149	500,302	Spoken language	Subtitles crawled from IMDb website
G1	105,341,070	392,635	Informative	News crawled from G1 news portal between 2014 and 2015.
PLN-Br [Bruckschen et al. 2008]	31,196,395	259,762	Informative	Large corpus of the PLN-BR Project with texts sampled from 1994 to 2005. It was also used by [Hartmann 2016] to train word embeddings models
Literacy works of public domain	23,750,521	381,697	Prose	A collection of 138,268 literary works from the Domínio Público website
Lacio-web [Aluísio et al. 2003]	8,962,718	196,077	Mixed genres	Texts from various genres, e.g., literary and its subdivisions (prose, poetry and drama), informative, scientific, law, didactic technical
Portuguese e-books	1,299,008	66,706	Prose	Collection of classical fiction books written in Brazilian Portuguese crawled from Literatura Brasileira website
Mundo Estranho	1,047,108	55,000	Informative	Texts crawled from Mundo Estranho magazine
CHC	941,032	36,522	Informative	Texts crawled from Ciência Hoje das Crianças (CHC) website
FAPESP	499,008	31,746	Science Communication	Brazilian science divulgation texts from Pesquisa FAPESP magazine
Textbooks	96,209	11,597	Didactic	Texts for children between 3rd and 7th-grade years of elementary school
Folhinha	73,575	9,207	Informative	News written for children, crawled in 2015 from Folhinha issue of Folha de São Paulo newspaper
NILC subcorpus	32,868	4,064	Informative	Texts written for children of 3rd and 4th-years of elementary school
Para Seu Filho Ler	21,224	3,942	Informative	News written for children, from Zero Hora newspaper
SARESP	13,308	3,293	Didactic	Text questions of Mathematics, Human Sciences, Nature Sciences and essay writing to evaluate students
Total	1,395,926,282	3,827,725		

Figura 3.1: Corpus utilizado na estimação dos Word Embeddings NILC [5]

.

3.3 Avaliação PMI e NPMI

Foi desenvolvida uma ferramenta para avaliação de tópicos utilizando as medidas PMI e NPMI. Essa ferramenta foi aplicada nesse trabalho para a comparação com a técnica proposta. Na ausência de uma biblioteca satisfatória que realizasse tal tarefa, uma API foi desenvolvida e disponibilizada[6]. Utilizando algumas bibliotecas *Python*, como *scipy* e *numpy*, a ferramenta desenvolvida calcula e armazena avaliações de tópicos a partir da busca de co-ocorrência das palavras dentro da Wikipedia em português.

Para avaliar um tópico qualquer pelas técnicas PMI e NPMI, é calculada a mediana da coerência de todos os pares de palavras daquele conjunto de termos, seguindo o especificado no trabalho *'External Evaluation of Topic Models'* [10]. Logo, considerando $\mathbf{w}$ o tópico a ser avaliado:

$$PMI\text{-}topico(\mathbf{w}) = mediana\{PMI(w_i, w_j) : w_i, w_j \in \mathbf{w}, i \neq j\}$$

[6]https://github.com/siqueiralex/topic-coherence

$$NPMI\text{-}topico(\mathbf{w}) = mediana\{NPMI(w_i, w_j) : w_i, w_j \in \mathbf{w}, i \neq j\}$$

Os cálculos PMI e NPMI são feitos de acordo com o especificado no referencial teórico deste trabalho. Os pares de palavras são procurados dentro de uma janela deslizante de 10 (dez) palavras. Este valor foi escolhido por ser o mais comum na literatura [11] [10] [1]. Entretanto, na ferramenta desenvolvida este parâmetro é ajustável.

3.3.1 Corpus e pré-processamento

Como base de conhecimento para a ferramenta de avaliação de tópicos, foi utilizado o dataset da Wikipedia em português obtido da Wikimedia Dumps[7] formado por 932.010 verbetes desta enciclopédia virtual. Seu pré-processamento consistiu na conversão de todos os caracteres para caixa-baixa (normalização), remoção de palavras de baixo valor semântico (stopwords) e retirada de caracteres numéricos e símbolos, deixando apenas o conteúdo estritamente textual dos verbetes. A escolha do dataset se deve aos ótimos resultados apresentados em trabalhos anteriores, que também utilizaram a Wikipedia como base de conhecimento para avaliacão automática de tópicos [10] [11] [1].

3.3.2 Arquitetura da Ferramenta

A figura 3.2 descreve a arquitetura da API desenvolvida. O *framework Django* foi escolhido para fornecer todas as estruturas *WEB* necessárias para o servidor. No diagrama, o módulo **Django Resquest Handler** é responsável por receber as requisições, identificar os *endpoints* e verificar se a requisição está de acordo com o especificado, respondendo com erro caso a requisição não esteja adequada. Este mesmo módulo aciona os demais que tratam de rotinas específicas de avaliação de tópicos. O funcionamento destes módulos será descrito a seguir:

Avaliador de tópicos

Módulo que recebe os tópicos passados pelo usuário e calcula sua coerência. Para fazer este cálculo, são utilizadas rotinas dos dois módulos subjacentes: **Calculador de co-ocorrência** e **Computador de coerência**. Os tópicos passados são avaliados e o resultado é devolvido ao usuário em um objeto do tipo *json*.

Calculador de co-ocorrência

Este módulo efetivamente calcula as ocorrências dos termos dentro da janela deslizante no corpus da Wikipedia. Uma rotina multithread é empregada para calcular o número de

[7] https://dumps.wikimedia.org/ptwiki/

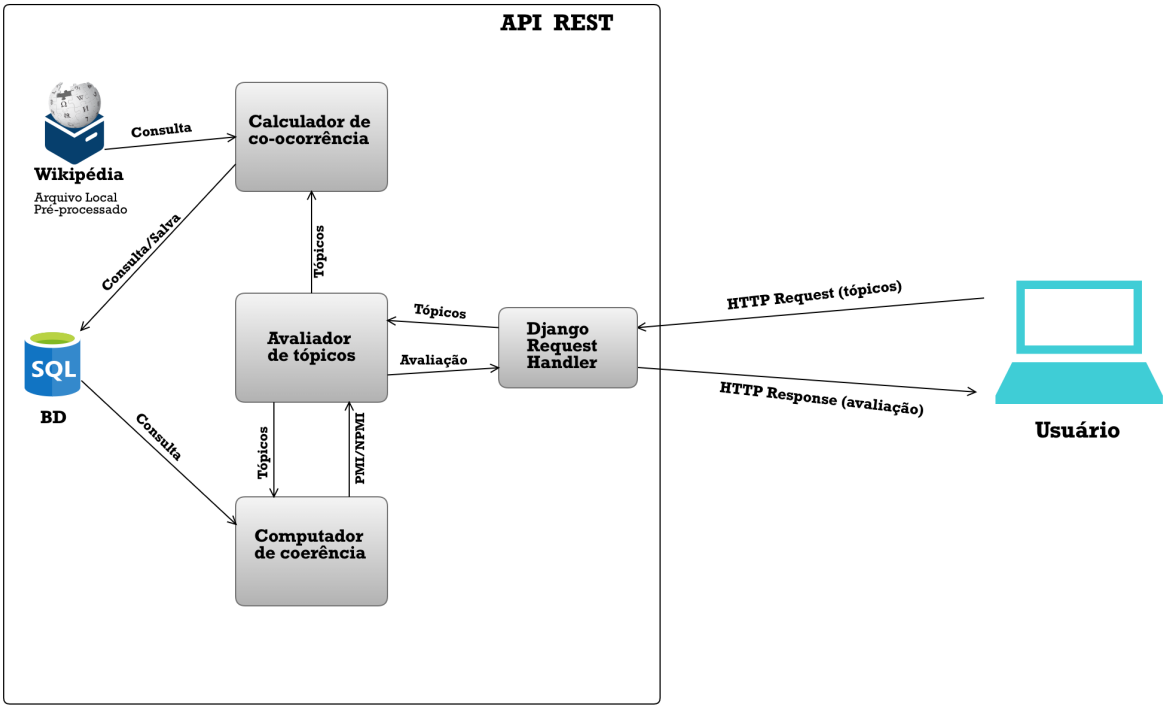

Figura 3.2: Arquitetura da API desenvolvida

vezes que cada par de palavras aparece dentro da janela, bem como a quantidade de vezes que cada palavra individualmente aparece. Esta rotina baseia-se no código[8] utilizado no artigo *"Machine Reading Tea Leaves: Automatically Evaluating Topic Coherence and Topic Model Quality"* [1]. O diferencial deste trabalho, entretanto, é a utilização de um Banco de Dados (MySQL) para armazenamento dos cálculos evitando assim a repetição da tarefa toda vez que um tópico é requisitado. O cálculo que este módulo realiza segue a lógica demonstrada no pseudocódigo à seguir:

```
1: procedure CONTAR_OCORRÊNCIAS(tópico)
2:     pares ← Dividir_Em_Pares(tópico)
3:     for par in pares do
4:         if par in Banco.todos_pares() then
5:             Remove(par,pares)
6:     for par in pares do
7:         ocorrências ← conta_ocorrências(par)
8:         Banco.salva(par, ocorrências)
```

[8]https://github.com/jhlau/topic_interpretability

Computador de coerência

Assumindo que a ocorrência das palavras já foram computadas e armazenadas no banco pelo processo anterior, o computador de coerência se utiliza destas informações para calcular a coerência propriamente dita.

Como foi especificado anteriormente, a coerência dos tópicos consiste na mediana da coerência de todos os pares de palavras. Observando as equações do PMI e NPMI para pares de palavras, nota-se que algumas probabilidades devem ser estimadas:

$$PMI(w_i, w_j) = \log \frac{P(w_i, w_j)}{P(w_i)P(w_j)}$$

$$NPMI(w_i, w_j) = \frac{PMI(w_i, w_j)}{-\log P(w_i, w_j)}$$

Para cada par de palavras w_i e w_j, $P(w_i, w_j)$, $P(w_i)$ e $P(w_j)$ precisam ser calculados. Isto é feito utilizando a quantidade de vezes que cada palavra aparece dentro da janela deslizante, bem como a quantidade de vezes que ambas apareceram juntas. Logo, considerando que w pode ser tanto uma palavra quanto um par de palavras, N_w a quantidade de vezes que w apareceu dentro de uma janela e $N_{janelas}$ a quantidade total de janelas observadas, a probabilidade de ocorrência de w é descrita pela equação:

$$p(w) = \frac{N_w}{N_{janelas}}$$

Como é garantido que N_w e $N_{janelas}$ estão salvos no banco, este cálculo é feito diretamente.

Portanto, este módulo computa a coerência de um tópico seguindo o pseudocódigo:

```
1: procedure CALCULAR_COERÊNCIA(tópico)
2:     pares ← Dividir_Em_Pares(tópico)
3:     coerencias ← [ ]
4:     for par in pares do
5:         ocorrências ← Banco.consulta(par)
6:         coerências.inclui( Calcula_Coerência(ocorrências) )
7:     return medianda(coerencias)
```

3.3.3 Utilização da Ferramenta

A ferramenta desenvolvida consiste numa API que recebe os tópicos do usuário e responde com seu coeficiente de avaliação PMI e NPMI. Isto é feito utilizando dois end-points distintos:

Endpoint '/api/topic/'

Este endpoint avalia apenas um tópico pelas métricas PMI e NPMI. Recebe uma requisição HTTP do tipo POST, que necessariamente deve conter um campo no seu corpo de nome 'topic', cujo conteúdo é do tipo textual e representa um tópico com seus termos separados por espaço. A resposta da API para este endpoint é um json com dois campos: 'pmi' e 'npmi', contendo os respectivos valores de coerência do tópico. A Figura 3.3 descreve um caso de uso deste endpoint no software *Postman*[9].

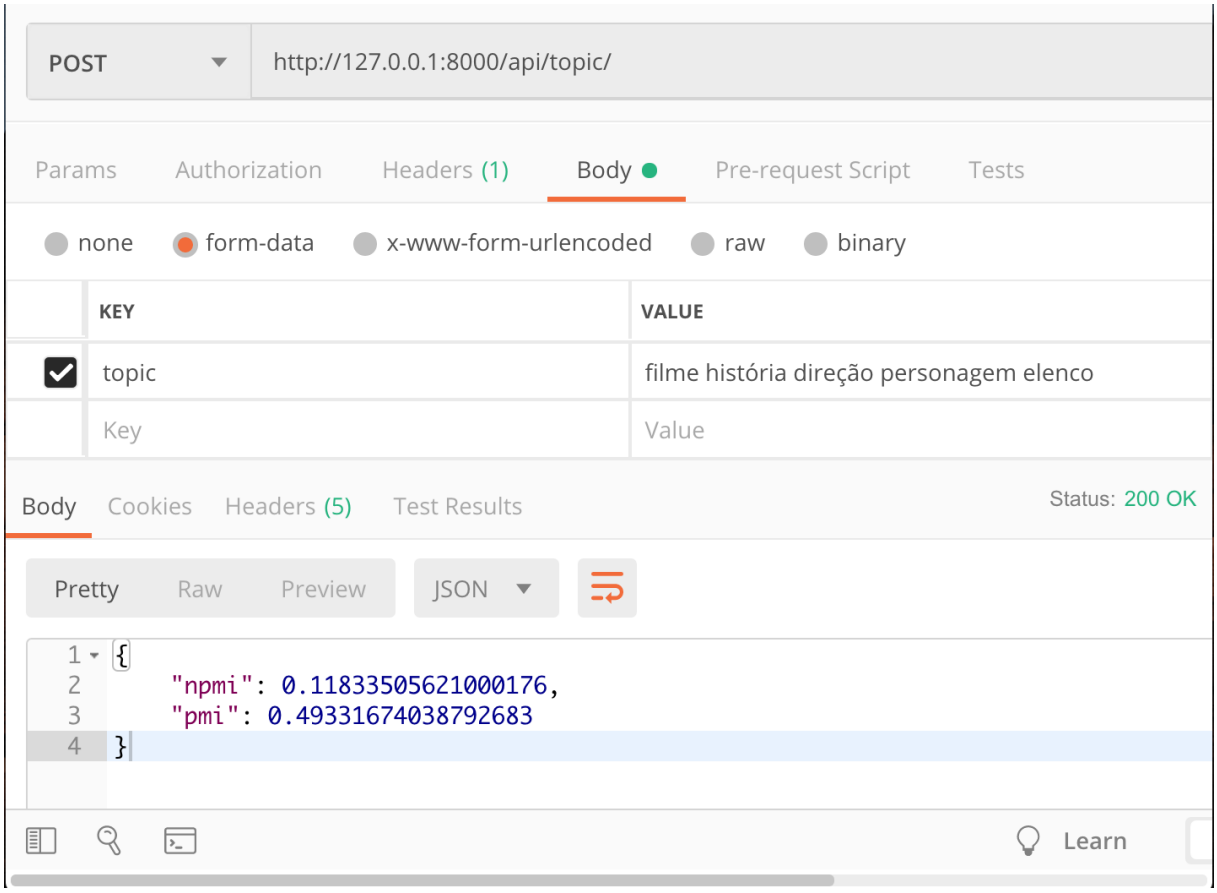

Figura 3.3: Caso de uso do endpoint '/api/topic/'

[9]https://www.getpostman.com/

Endpoint '/api/model/'

Este endpoint foi desenvolvido para avaliar múltiplos tópicos ao mesmo tempo e responder com as coerências individuais bem como a média e mediana de todos os tópicos enviados pelo usuário. Portanto, esta funcionalidade é útil para comparar vários modelos de tópicos a partir da totalidade dos tópicos gerados.

Recebe requisições HTTP do tipo POST, que necessariamente devem conter um json com o campo 'topics' cujo conteúdo é uma lista de tópicos, cada um representado por palavras separadas por espaços. A resposta da API para este endpoint consiste em um json com dois campos: 'pmi' e 'npmi', cada um contendo um objeto consistindo nas avaliações individuais de todos os tópicos, bem como a média e a mediana dos valores encontrados. A figura 3.3 descreve um caso de uso deste endpoint no software *Postman*.

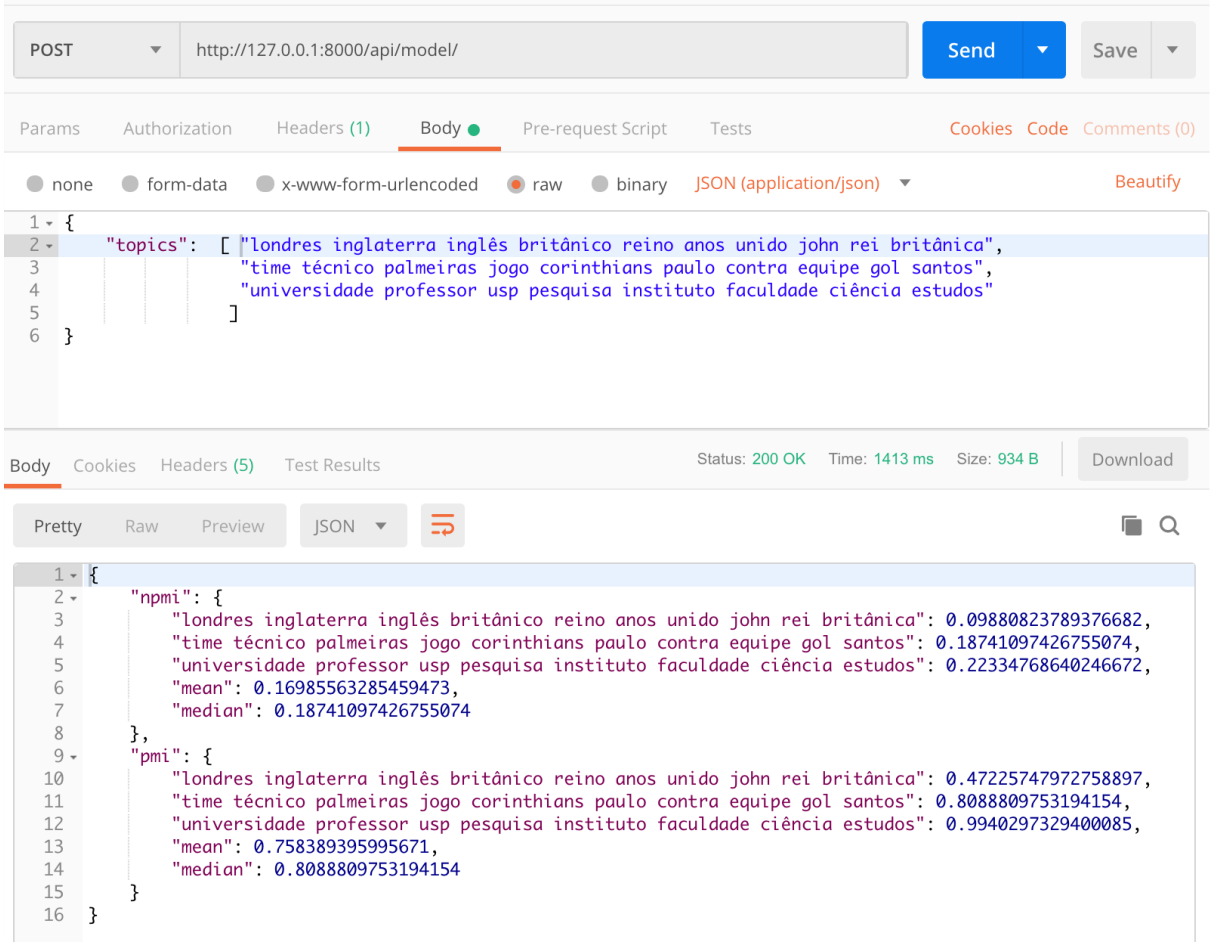

Figura 3.4: Caso de uso do endpoint '/api/model/'

3.4 Avaliação de Tópicos por Vetores de Palavras

Para avaliar tópicos a partir dos vetores de palavras estimados, em analogia com o que é feito na avaliação por PMI e NPMI, foi utilizada a mediana da similaridade de todos os pares de palavras presentes no tópico. Para calcular a similaridade entre pares de palavras, é calculada a medida de similaridade cosseno (ou distância cosseno), que é basicamente o cálculo do cosseno entre os dois vetores. Esta técnica é a mais comum para se avaliar similaridade entre termos, sendo utilizada em vários trabalhos acadêmicos que lidam com word embedding [3] [4] [26].

$$Avaliacao\text{-}topico(\mathbf{w}) = mediana\{Similaridade(w_i, w_j) : w_i, w_j \in \mathbf{w}, i \neq j\}$$

$$Similaridade(w_i, w_j) = \frac{w_i \cdot w_j}{\|w_i\| \, \|w_j\|}$$

Também em analogia com o que é feito nas técnicas PMI e NPMI, para se avaliar um modelo utilizando word embedding, foi utilizada a mediana da avaliação de todos os tópicos gerados por ele.

3.5 Comparações

Como foi descrito no referencial teórico deste trabalho, a avaliação da coerência semântica dos termos que formam tópicos surgiu como alternativa à avaliação de modelos probabilísticos de tópicos [9]. Portanto, este tipo de análise serve como forma objetiva de identificar os 'melhores' e 'piores' tópicos de um modelo, mas também, ao se analisar conjuntos de tópicos gerados por modelos, se faz possível graduar e ranquear vários modelos independentemente do algoritmo utilizado para gerar seus tópicos. Desta forma, seguindo o formato utilizado no artigo *'Machine Reading Tea Leaves: Automatically Evaluating Topic Coherence and Topic Model Quality'* [1], este trabalho separa as comparações em dois grupos: a correlação na avaliação de tópicos e a correlação na avaliacão de modelos (conjuntos de tópicos).

Para cada um dos modelos gerados neste trabalho, será calculada a correlação na avaliação dos seus tópicos. Ou seja, todos os tópicos de cada modelo serão avaliados pelas técnicas que utilizam word embedding e pelas técnicas PMI e NPMI, por fim as correlações destas medidas serão estimadas utilizando o coeficiente *Pearson*.

As técnicas de avaliação nesta etapa serão utilizadas para avaliar modelos (conjuntos de tópicos). Para tal, todos os modelos de tópicos gerados neste trabalho serão avaliados por ambas técnicas, tendo como medida de coerência a mediana da avaliação de todos os

tópicos. Por fim, a correlação das avaliações utilizando a técnica de word embedding e as técnicas PMI e NPMI serão estimadas utilizando o coeficiente *Pearson*.

Capítulo 4

Resultados

Nesta seção serão expostos e discutidos os resultados dos experimentos realizados. Neles foram utilizados vetores de palavras extraídos usando a técnica *Word2Vec* [3] em português para avaliar modelos de tópicos. Os resultados foram comparados com as técnicas PMI e NPMI por serem aquelas com os melhores resultados na literatura [1, 11].

Os gráficos gerados descrevem a correlação das avaliações em função das dimensões dos vetores de palavras. Por ser um parâmetro muito importante, que interfere significativamente no custo computacional da estimação dos vetores, julgou-se importante saber com quantas dimensões os vetores atingem avaliações de tópicos eficientes.

4.1 Avaliação de Tópicos

Todos os tópicos gerados foram avaliados aplicando o cálculo do PMI e NPMI. Depois, os mesmo tópicos foram avaliados pelos vetores das palavras. As correlações destas medidas foram calculadas utilizando o coeficiente de Pearson (Pearson-r).

Na Figura 4.1, estão os gráficos das correlações Pearson pelo número de dimensões dos vetores de palavras em comparação com avaliação PMI. Observa-se que os vetores estimados utilizando a técnica Skip-Gram alcançaram valores de correlação maiores que os estimados por Continuous Bag-of-word, especialmente em baixas dimensões. Os vetores treinados na Wikipedia (Word2Vec-Wiki Skip-Gram) apresentaram máxima correlação de 0.7 em 300 dimensões. Já os vetores treinados pelo NILC (Word2Vec-NILC Skip-Gram) apresentaram máxima correlação de 0.72 em 100 dimensões.

Os vetores do tipo Continuous Bag-of-words apresentaram comportamento instável, aqueles treinados pelo NILC (Word2Vec-NILC CBOW) apresentaram máxima correlação de 0.67 em 100 dimensões, e para os estimados na Wikipedia (Word2Vec-Wiki CBOW), observou-se correlação máxima de 0.7 em 1000 dimensões.

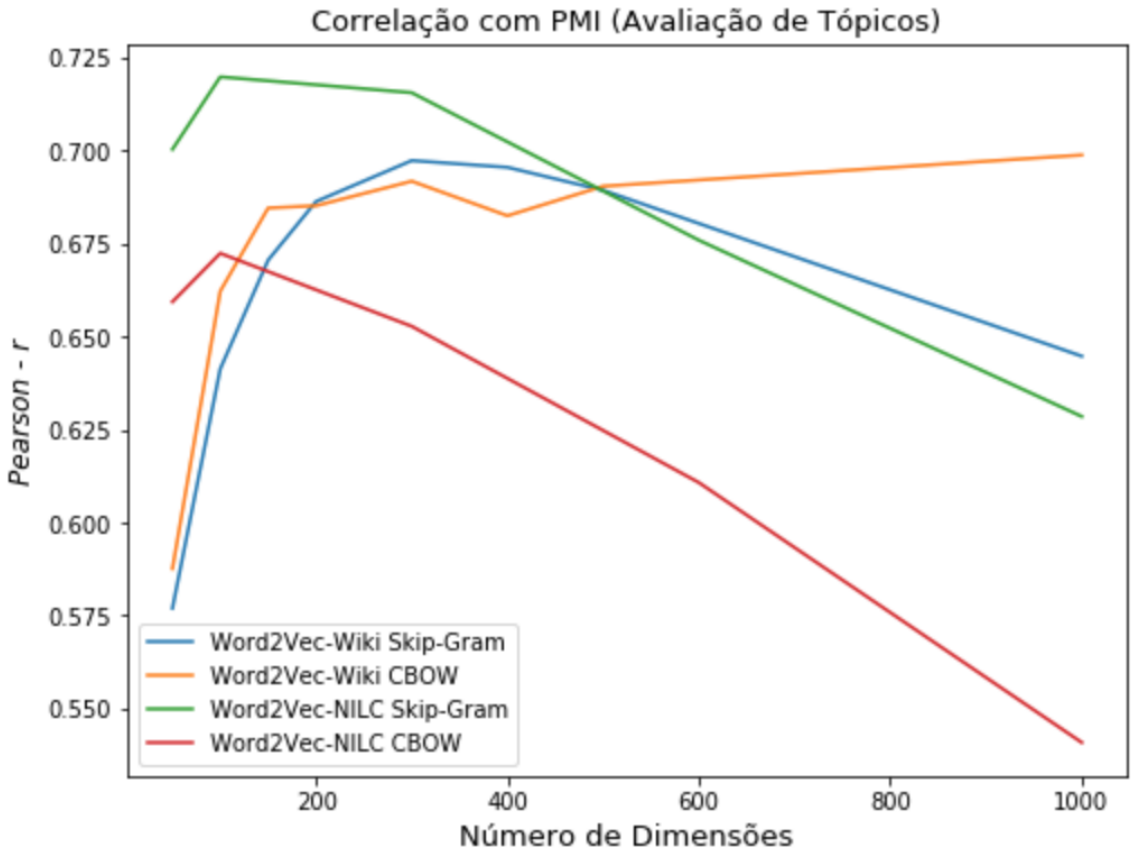

Figura 4.1: Correlação com PMI na avaliação de tópicos.

Na Figura 4.2 observou-se um padrão parecido com a figura anterior, nela estão plotadas as correlações dos vetores com as avaliações NPMI em funcão da dimensão. Novamente os vetores do tipo Skip-Gram tiveram desempenho superior especialmente em baixas dimensões, aqueles treinados pelo NILC (Word2vec-NILC Skip-Gram) apresentaram correlação máxima de 0.76 em 300 dimensões. Já aqueles treinados na Wikipedia (Word2Vec-Wiki Skip-Gram) apresentaram correlação máxima de 0.74 em 400 dimensões.

Os vetores do tipo Continuous Bag-of-words novamente apresentaram comportamento instável. Aqueles treinados pelo NILC (Word2Vec-Wiki NILC) apresentaram correlação máxima de 0.69 em 100 dimensões e aqueles treinados na Wikipedia (Word2Vec-Wiki CBOW) apresentaram correlação máxima de 0.74 em 1000 dimensões.

Com base nestas observações, foi realizado um experimento variando o tamanho da janela de contexto, que é o número de palavras 'vizinhas' utilizadas para estimar os vetores. Julgando que os vetores do tipo Skip-gram obtiveram melhores resultados, foi escolhida esta técnica e determinado o valor de 300 para fixar o número de dimensões, por ser aquele em que os vetores treinados na Wikipedia apresentaram melhores resultados.

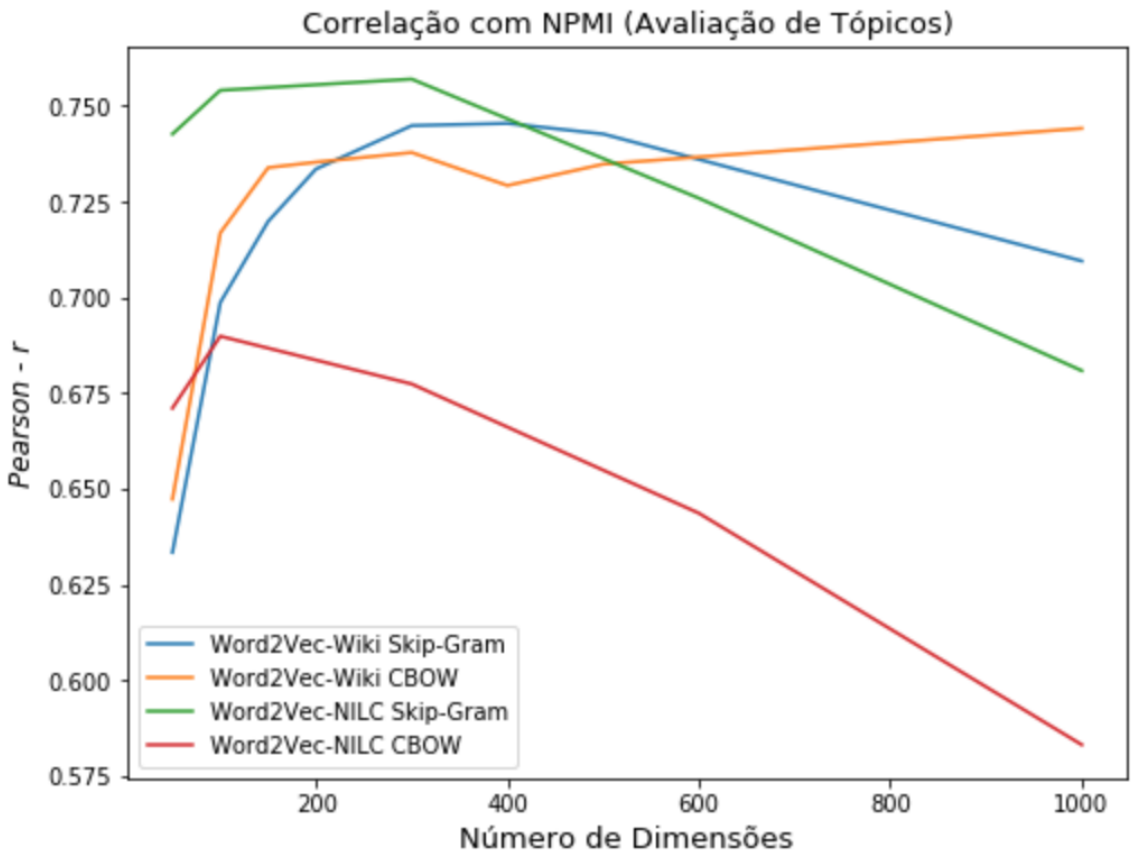

Figura 4.2: Correlação com NPMI na avaliação de tópicos.

Os resultados de correlação com PMI e NPMI podem ser observados na Figura 4.3, em que se observou que os vetores treinados utilizando janelas de 6 palavras apresentaram os melhores resultados, atingindo correlação de 0.72 com PMI e 0.77 com NPMI.

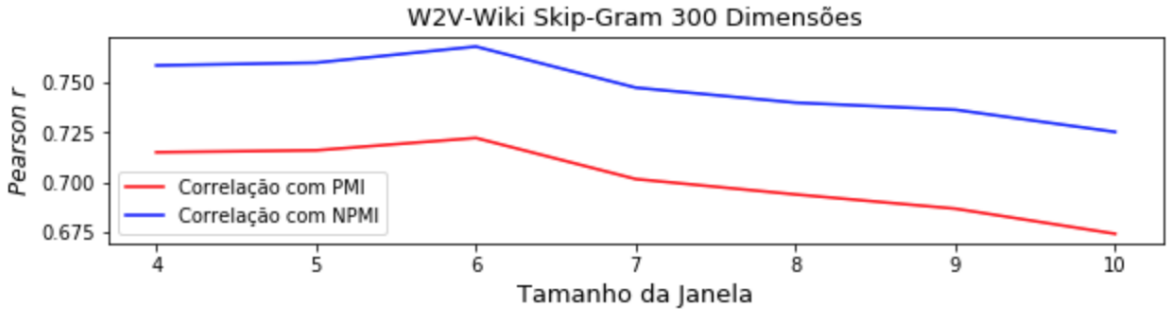

Figura 4.3: Correlação com PMI/NPMI em função da janela de contexto

4.1.1 Discussão

De maneira geral, os resultados observados demonstraram correlações razoavelmente grandes entre as avaliações por vetores de palavras e avaliações por PMI e NPMI. Notadamente, outros estudos presentes no referencial teórico deste trabalho consideraram 'altas' correlações acima de 0.7 [10] [1]. Além disso, é esperado que se observe algum grau de correlação entre diferentes métodos de avaliação que utilizam a mesma fonte de conhecimento (Wikipedia), e esta intuição foi corroborada pelos resultados obtidos.

Para todos os vetores analisados, a correlação com NPMI foi sempre maior que com PMI. Também, apesar das correlações observadas serem similares, vetores obtidos pela técnica Skip-Gram apresentaram melhores resultados que os obtidos por Continuous Bag-of-Words, visto que apresentaram maiores correlações em menos dimensões e, portanto, menor custo computacional. Isto foi observado tanto para os vetores treinados exclusivamente na Wikipedia quanto para os vetores NILC, pré-treinados em corpus bem maiores.

Avaliando os resultados de todos os vetores utilizados, aqueles treinados na Wikipedia pela técnica Skip-Gram com 300 dimensões e janela de contexto tamanho 6 foram os que obtiveram melhores resultados, seguidos pelos vetores NILC pré-treinados através da técnica Skip-Gram com 100 dimensões. Entretanto, os vetores NILC se destacam com resultados melhores para baixas dimensões, com correlações maiores que 0.7 para vetores de apenas 50 dimensões, demonstrando a qualidade obtida ao se treinar vetores em uma quantidade muito maior de documentos das mais variadas áreas.

Selecionamos os 8 tópicos mais bem avaliados pelas técnicas PMI e Skip-Gram (treinado pelo NILC) na tabela 4.1, observando as 4 palavras mais relevantes para cada tópico, fica evidente as principais diferenças entre as duas técnicas.

Enquanto os tópicos mais bem avaliados por PMI têm fácil interpretação como assuntos reais, por exemplo o tópico 'senna piloto carro corrida' que claramente reflete o assunto 'Ayrton Senna', os tópicos mais bem avaliados por Skip-gram agrupam palavras com as mesmas características, mas que não necessariamente formam um tópico que reflete um assunto real. O tópico 'dia dias mês maio', por exemplo, é formado por substantivos que indicam informação de tempo, porém, por mais que essas palavras sintaticamente tenham função parecida, não parecem agrupar logicamente documentos e portanto não formam um tópico de boa interpretabilidade. Outro exemplo é o tópico 'josé carlos silva luiz', todas as palavras formadoras são nomes próprios, porém, não são um bom conjunto de palavras para expressar um assunto/tópico. Desta forma fica evidenciado que os vetores de palavras, por trabalharem com aproximação por contexto, podem em alguns momentos agrupar palavras que têm a mesma função sintática mas que, mesmo assim, não formam tópicos de boa interpretabilidade. Enquanto as técnicas NPMI/PMI não parece ter o

Tabela 4.1: Tópicos mais bem avaliados por PMI e Skip-Gram..

PMI	**NILC Skip-Gram**
‘israel paz israelense palestinos’	‘josé carlos silva luiz’
‘restaurante cozinha pratos comida’	‘dia dias mês maio’
‘partido candidato eleitoral eleições’	‘tv rádio programa televisão’
‘imposto sobre receita impostos federal’	‘feira semana segunda sexta’
‘senna piloto carro corrida’	‘pfl psdb pmdb governo’
‘presidente fernando henrique itamar’	‘banda rock disco música’
‘cor cores olhos azul’	‘doença aids vírus casos’
‘candidato psdb quércia campanha’	‘cor cores olhos azul’

mesmo problema. Os conjuntos de tópicos mais bem avaliados por cada uma das técnicas estão integralmente disponíveis nos anexos deste trabalho.

4.2 Avaliação de Modelos

Na segunda parte dos experimentos, observamos as correlações na avaliação de modelos, considerando como coerência de modelo a mediana da avaliação de todos os seus tópicos.

Inicialmente, foram calculadas as correlações das avaliações por vetores de palavras com a técnica PMI em função da dimensão, os resultados estão na Figura 4.4. Aqueles vetores que utilizaram a técnica Skip-Gram treinados pelo NILC (Word2Vec-NILC Skip-Gram) atingiram a maior correlação: 0.93 em 600 dimensões. Já as correlações observadas utilizando vetores treinados na Wikipedia pela técnica Skip-Gram (Word2Vec-Wiki Skip-Gram) apresentaram correlação máxima de 0.6 em 500 dimensões. Em geral, os vetores treinados por Continuous Bag-of-words apresentaram correlações baixas, aqueles treinados na wikipedia (Word2Vec-Wiki CBOW) apresentaram valores negativos em todo o espectro, enquanto aqueles treinados pelo NILC (Word2Vec-NILC CBOW), mostraram correlação máxima de 0.63 em 100 dimensões.

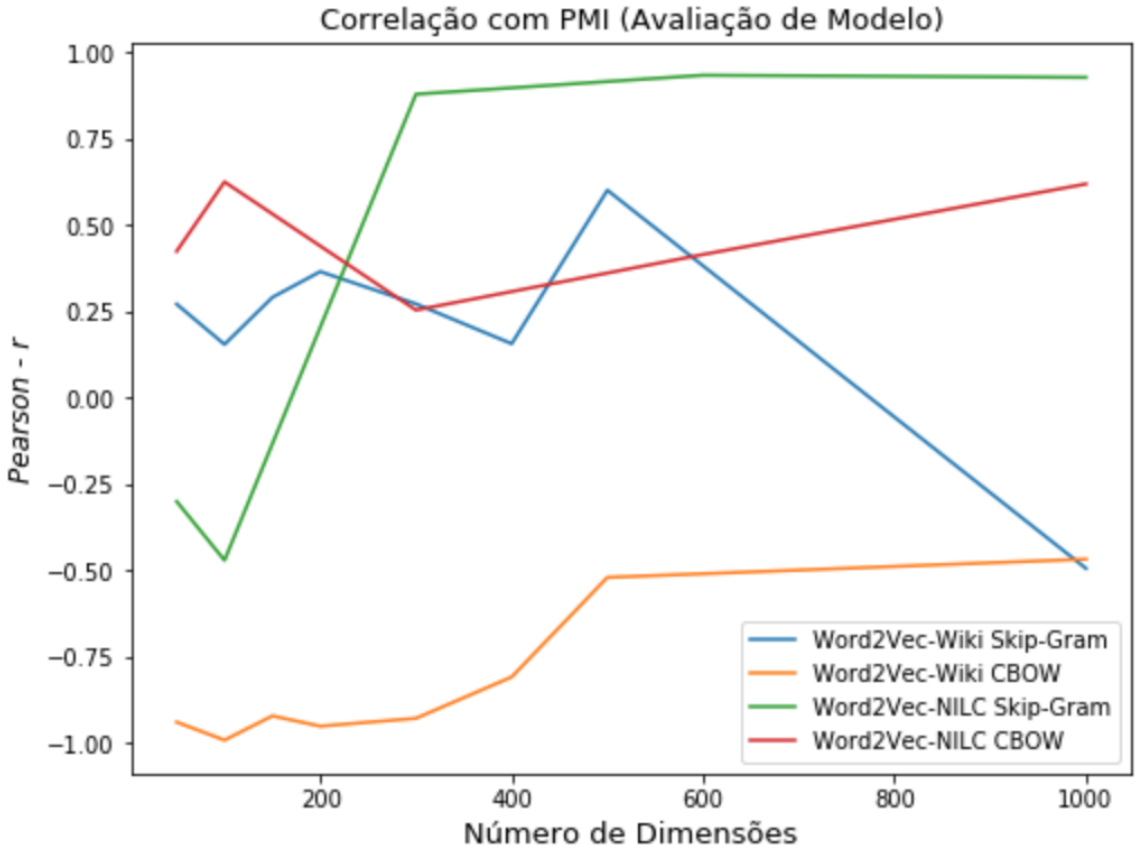

Figura 4.4: Correlação com PMI na avaliacão de modelos.

Observando as correlações com NPMI, presentes na Figura 4.5, observa-se novamente

que os vetores estimados por Skip-Gram apresentam maior correlação. Aqueles treinados na Wikipedia (Word2Vec-Wiki Skip-Gram) apresentaram correlação máxima de 0.98 em 500 dimensões. Os vetores treinados pelo NILC (Word2Vec-NILC Skip-Gram) apresentaram correlação máxima de 0.92 em 1000 dimensões.

Novamente, vetores estimados por Continuous Bag-of-words apresentaram majoritariamente valores negativos, à exceção daqueles treinados na Wikipedia que apresentaram valores ligeiramente acima de zero em 1000 dimensões.

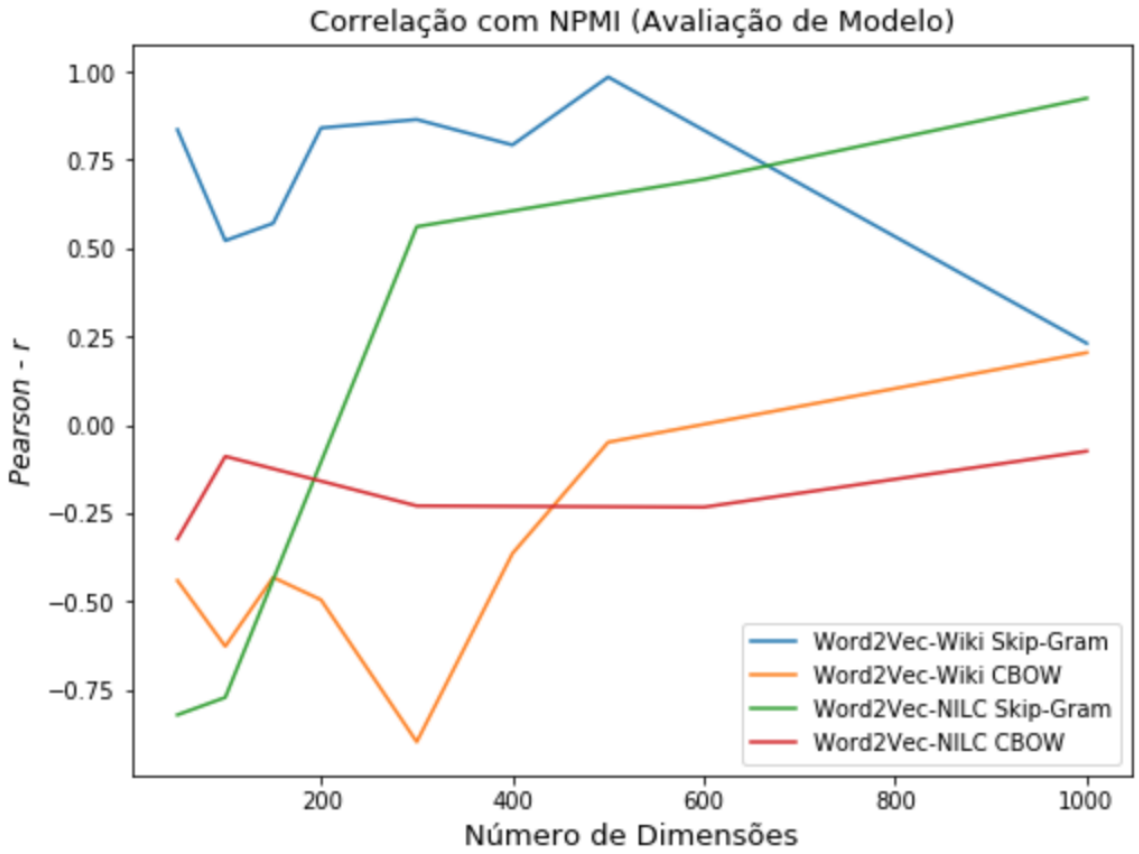

Figura 4.5: Correlação com NPMI na avaliacão de modelos.

4.2.1 Discussão

Para avaliação de modelos, as correlações observadas foram em geral maiores que na avaliação de tópicos. Este padrão também foi observado no trabalho *'Machine Reading Tea Leaves: Automatically Evaluating Topic Coherence and Topic Model Quality'*, porém, neste se avaliou a correlação do PMI e NPMI com anotadores humanos.

Novamente, os vetores disponibilizados pelo NILC apresentaram resultados mais consistentes que os treinados apenas na Wikipedia. Esses vetores, treinados em bases de

conhecimento bem maiores, atingiram valores de correlação satisfatórios a partir de 300 dimensões (0.88 com PMI e 0.56 com NPMI) e valores máximos de correlação bastante altos (0.93 com PMI e 0.92 com NPMI). Portanto, com vetores NILC apresentando os melhores resultados e correlações mais consistentes em toda faixa de dimensões, estes experimentos demonstraram que vetores de palavras estimados em bases de conhecimentos maiores e de domínios variados adquirem qualidade maior que aqueles treinados em bases menores (à exemplo dos treinados apenas na Wikipedia). Isto foi arguido no trabalho que disponibilizou estes vetores *'Portuguese Word Embeddings: Evaluating on Word Analogies and Natural Language Tasks'* [5].

Assim como na avaliação de tópicos, ficou evidenciado que vetores treinados utilizando a técnica Skip-Gram são mais adequados para avaliar tópicos com o objetivo de emular as avaliações PMI e NPMI, atingindo correlações acima de 0.9 em alguns casos. Em contraste, os vetores estimados por Continuous Bag-of-words frequentemente apresentaram correlações negativas com NPMI e PMI. Com valores altos de correlação, julga-se então como promissora a aplicabilidade de vetores do tipo Skip-Gram treinados em grandes bases de conhecimento na avaliação de tópicos, necessitando de trabalhos futuros mais aprofundados para validar tal aplicação na prática.

Capítulo 5

Conclusão

Este trabalho investigou a aplicabilidade de vetores de palavras na tarefa de avaliação de tópicos. Isto foi feito comparando-se os resultados com as duas técnicas mais bem estabelecidas na literatura: PMI e NPMI. Ambas se baseiam no cálculo da probabilidade de co-ocorrência de pares de palavras em grandes bases de conhecimento, mais comumente a Wikipedia. Portanto, apesar dos bons resultados apresentados, PMI e NPMI são técnicas custosas computacionalmente.

A técnica proposta é, então, uma alternativa muito mais rápida, visto que o gasto computacional é empregado majoritariamente em tempo de estimação dos vetores de palavras. Para realizar avaliação de tópicos, apenas operações algébricas entre vetores são realizadas.

A intuição por trás desta investigação vem de vários pontos de intersecção entre ambas áreas que sugerem que esta aplicação seja possível. Primeiro, a utilização da bases de conhecimento para definir relações semânticas entre palavras, com a Wikipedia sendo escolhida para as rotinas desenvolvidas neste trabalho. Segundo, o conceito de janela de contexto utilizada no Word2Vec, que se assemelha à janela deslizante utilizada no PMI/NPMI.

Ressalta-se que algumas diferenças entre as técnicas já sugeriam que a correlação não poderia ser completa. Por exemplo, o fato de que o Word2Vec chega a estabelecer relações semânticas complexas entre termos (ex: país-capital), enquanto PMI/NPMI, por definição, apenas exprime a relação de proximidade entre termos na base de conhecimento. Também, como ficou demonstrado, Word2Vec tende a aproximar palavras com funções sintáticas parecidas mas que não necessariamente formam assuntos interpretáveis como reais. Isto pode ser observado analisando os tópicos mais bem avaliados por cada técnica nos anexos deste trabalho.

Desta forma, considera-se que os resultados apresentados neste trabalho demonstram como promissora a aplicabilidade de vetores de palavras em avaliação de tópicos, mais

especificamente aqueles estimados pela técnica Word2Vec (Skip-Gram) em grandes bases de arquivos textuais, como aqueles disponibilizados pelo Núcleo Interinstitucional de Linguística Computacional-NILC/USP (NILC), que apresentaram correlações altas e consistentes nas tarefas de avaliação de tópicos e de modelos.

5.1 Trabalhos futuros

Para confirmar a eficiência desta aplicação, julga-se conveniente trabalhos mais abrangentes que avaliem e comparem técnicas de word embedding para avaliação de tópicos com notas dadas por especialistas humanos, tanto na avaliação direta como nas tarefas de intrusão de palavras e de tópicos. Outro caminho interessante seria analisar os efeitos da aplicação desta técnica nos modelos, em estudos quanto a melhoria da performance e otimização semântica de modelos que utilizem avaliações por word embedding para melhorar sua qualidade interna em tempo de treinamento.

Referências

[1] Lau, Jey Han e David Newman: *Machine Reading Tea Leaves: Automatically Evaluating Topic Coherence and Topic Model Quality*. abril 2014. ix, 3, 10, 11, 21, 25, 26, 30, 32, 35

[2] Bengio, Yoshua, Réjean Ducharme, Pascal Vincent e Christian Janvin: *A Neural Probabilistic Language Model*. J. Mach. Learn. Res., 3:1137–1155, março 2003, ISSN 1532-4435. `http://dl.acm.org/citation.cfm?id=944919.944966`. ix, 2, 11, 15, 16

[3] Mikolov, Tomas, Kai Chen, Greg Corrado e Jeffrey Dean: *Efficient Estimation of Word Representations in Vector Space*. Em *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013. `http://arxiv.org/abs/1301.3781`. ix, 2, 17, 21, 22, 30, 32

[4] Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg Corrado e Jeffrey Dean: *Distributed Representations of Words and Phrases and Their Compositionality*. Em *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, páginas 3111–3119, USA, 2013. Curran Associates Inc. `http://dl.acm.org/citation.cfm?id=2999792.2999959`, event-place: Lake Tahoe, Nevada. ix, 17, 18, 19, 30

[5] Hartmann, Nathan, Erick Fonseca, Christopher Shulby, Marcos Treviso, Jessica Rodrigues e Sandra Aluisio: *Portuguese Word Embeddings: Evaluating on Word Analogies and Natural Language Tasks*. arXiv:1708.06025 [cs], agosto 2017. `http://arxiv.org/abs/1708.06025`, acesso em 2019-04-11, arXiv: 1708.06025. ix, 23, 24, 39

[6] Blei, David M.: *Introduction to Probabilistic Topic Models*. Communications of the ACM, 55, 2011. 1, 4, 5, 6, 22

[7] Mimno, David e Wallach Hanna M.: *Optimizing Semantic Coherence in Topic Models*. EMNLP '11, 2011. `http://dl.acm.org/citation.cfm?id=2145432.2145462`. 2

[8] Wallach, Hanna M., Iain Murray, Ruslan Salakhutdinov e David Mimno: *Evaluation Methods for Topic Models*. Em *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, páginas 1105–1112, New York, NY, USA, 2009. ACM, ISBN 978-1-60558-516-1. 2, 7

[9] Chang, Jonathan, Sean Gerrish, Chong Wang, Jordan L. Boyd-graber e David M. Blei: *Reading Tea Leaves: How Humans Interpret Topic Models*. páginas 288–296. Curran Associates, Inc., 2009. 2, 4, 7, 22, 30

[10] Newman, David, Sarvnaz Karimi e Lawrence Cavedon: *External Evaluation of Topic Models*. 2009. 2, 3, 7, 21, 24, 25, 35

[11] Newman, David, Jei Han Lau e Karl Grieser: *Automatic Evaluation of Topic Coherence*. 2010. 2, 9, 21, 25, 32

[12] Blei, David M., Andrew Y. Ng e Michael I. Jordan: *Latent Dirichlet Allocation*. J. Mach. Learn. Res., 3, 2003. 4, 5, 6, 21

[13] Princeton, New Jersey: *About Wordnet*. `https://wordnet.princeton.edu/`, acesso em 2019-05-15. 9

[14] *Wikipedia Estatísticas*. `https://pt.wikipedia.org/wiki/Wikip%C3%A9dia:Estat%C3%ADsticas`. 9

[15] Bouma, Gerlof: *Normalized (Pointwise) Mutual Information in Collocation Extraction*. Proceedings of the Biennial GSCL Conference 2009, 2009. 11

[16] Mikolov, Tomas: *Using Neural Networks for Modeling and Representing Natural Languages*. Em *COLING 2014, 25th International Conference on Computational Linguistics, Tutorial Abstracts, August 23-29, 2014, Dublin, Ireland*, páginas 3–4, 2014. `http://aclweb.org/anthology/C/C14/C14-3002.pdf`. 12

[17] Robertson, Stephen: *Understanding inverse document frequency: On theoretical arguments for IDF*. Journal of Documentation, 60:2004, 2004. 13

[18] Harris, Zellig S.: *Distributional Structure*. *WORD*, 10(2-3):146–162, agosto 1954, ISSN 0043-7956, 2373-5112. `http://www.tandfonline.com/doi/full/10.1080/00437956.1954.11659520`, acesso em 2019-05-20. 13

[19] Turney, Peter D. e Patrick Pantel: *From Frequency to Meaning: Vector Space Models of Semantics*. CoRR, abs/1003.1141, 2010. `http://arxiv.org/abs/1003.1141`. 14

[20] Stewart, G. W.: *On the Early History of the Singular Value Decomposition*. 1992. 14

[21] Aletras, Nikolaos e Mark Stevenson: *Evaluating Topic Coherence Using Distributional Semantics*. 2013. 14

[22] Mitchell, Thomas M.: *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1ª edição, 1997, ISBN 0-07-042807-7 978-0-07-042807-2. 16

[23] Kent State, University: *SPSS Tutorials: Pearson Correlation*, maio 2019. `https://libguides.library.kent.edu/SPSS/PearsonCorr`, acesso em 2019-05-28. 20, 21

[24] Stevens, Keith e Philip Kegelmeyer: *Exploring Topic Coherence over many models and many topics*. 2012. 22

[25] Bhatia, Shraey, Jey Han Lau e Timothy Baldwin: *An Automatic Approach for Document-level Topic Model Evaluation.* Em *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, páginas 206–215, Vancouver, Canada, agosto 2017. Association for Computational Linguistics. `https://www.aclweb.org/anthology/K17-1022`. 22

[26] Levy, Omer e Yoav Goldberg: *Neural Word Embedding As Implicit Matrix Factorization.* Em *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, páginas 2177–2185, Cambridge, MA, USA, 2014. MIT Press. `http://dl.acm.org/citation.cfm?id=2969033.2969070`, event-place: Montreal, Canada. 30

Anexo I

Lista de tópicos mais bem avaliados

I.1 Tópicos mais bem avaliados por PMI

1.93 - israel paz israelense palestinos acordo arafat palestina rabin olp israelenses
1.57 - restaurante cozinha pratos comida casa restaurantes carne comer cardápio prato
1.48 - partido candidato eleitoral eleições eleição psdb pmdb candidatos votos governador
1.48 - imposto sobre receita impostos federal tributária arrecadação fiscal renda icms
1.40 - senna piloto carro corrida equipe ayrton gp fórmula pilotos pista
1.34 - presidente fernando henrique itamar governo cardoso franco ministro fhc collor
1.34 - cor cores olhos azul cabelo branco pele cabelos vermelho verde
1.32 - candidato psdb quércia campanha pmdb fhc lula candidatura fernando henrique
1.31 - carro carros veículos fábrica ford veículo fiat motor modelo volkswagen
1.31 - ônibus transporte metrô paulo trem transportes linha estação passageiros trens
1.30 - eleitoral eleições eleição partido votos presidente voto candidatos partidos
1.29 - saúde hospital médico médicos hospitais atendimento medicina médica jatene
1.27 - tv rádio programa televisão globo rede canal programas emissoras emissora
1.25 - doença aids vírus casos saúde câncer sangue doenças tratamento hiv
1.20 - pfl psdb pmdb senador partido sarney magalhães presidente fhc antônio
1.18 - câmbio bilhões dólar comercial cambial importações mercado exportações dólares
1.16 - pt lula partido candidato campanha silva petista brizola luiz inácio
1.16 - deputado comissão câmara cpi deputados senador senado parlamentares brasília pmdb
1.14 - justiça direito lei tribunal federal judiciário stf supremo advogados juízes
1.10 - itália italiano italiana roma italianos berlusconi milan milão sílvio di

I.2 Tópicos mais bem avaliados por NPMI

0.32 - partido candidato eleitoral eleições eleição psdb pmdb candidatos votos governador
0.32 - israel paz israelense palestinos acordo arafat palestina rabin olp israelenses

0.32 - tv rádio programa televisão globo rede canal programas emissoras emissora
0.28 - cor cores olhos azul cabelo branco pele cabelos vermelho verde
0.28 - senna piloto carro corrida equipe ayrton gp fórmula pilotos pista
0.28 - imposto sobre receita impostos federal tributária arrecadação fiscal renda icms
0.27 - restaurante cozinha pratos comida casa restaurantes carne comer cardápio prato
0.27 - ônibus transporte metrô paulo trem transportes linha estação passageiros trens
0.26 - presidente fernando henrique itamar governo cardoso franco ministro fhc collor
0.26 - saúde hospital médico médicos hospitais atendimento medicina médica jatene
0.25 - carro carros veículos fábrica ford veículo fiat motor modelo volkswagen
0.25 - doença aids vírus casos saúde câncer sangue doenças tratamento hiv
0.24 - escolas escola educação alunos curso ensino professores aulas cursos grau
0.24 - deputado comissão câmara cpi deputados senador senado parlamentares brasília pmdb
0.24 - universidade escola educação escolas professor alunos usp curso professores ensino
0.23 - itália italiano italiana roma italianos berlusconi milan milão sílvio di
0.23 - justiça direito lei tribunal federal judiciário stf supremo advogados juízes
0.23 - josé carlos silva luiz joão roberto pereira lima alves santos
0.21 - música orquestra ópera compositor concerto piano musical maestro músicos teatro
0.21 - futebol clube paulo campeonato corinthians brasileiro jogo palmeiras paulista

I.3 Tópicos mais bem avaliados pelos vetores Skip-Gram - NILC

0.78 - josé carlos silva luiz joão roberto pereira lima alves santos
0.67 - dia dias mês maio abril julho março setembro dezembro agosto
0.59 - tv rádio programa televisão globo rede canal programas emissoras emissora
0.57 - feira semana segunda sexta última quarta quinta terça passada próxima
0.57 - pfl psdb pmdb governo deputado senador partido líder presidente magalhães
0.57 - banda rock disco música grupo bandas pop rap show som
0.54 - doença aids vírus casos saúde câncer sangue doenças tratamento hiv
0.54 - cor cores olhos azul cabelo branco pele cabelos vermelho verde
0.51 - universidade escola educação escolas professor alunos usp curso professores ensino
0.51 - presidente fernando henrique itamar governo cardoso franco ministro fhc collor
0.50 - carro carros veículos fábrica ford veículo fiat motor modelo volkswagen
0.48 - avião aeroporto viagem vôo passageiros trem navio aviões transporte carga
0.47 - fez ficou chegou conseguiu acabou começou deu passou antes deixou
0.47 - poeta literatura escritor poesia autor obra romance livro vida texto
0.46 - restaurante cozinha pratos comida casa restaurantes cardápio prato molho vinho
0.45 - justiça direito lei tribunal federal judiciário stf supremo advogados juízes
0.45 - arte exposição museu artista artistas obras mostra bienal obra pintura

0.45 - ônibus transporte metrô paulo trem transportes linha estação passageiros trens
0.44 - israel paz israelense palestinos acordo arafat palestina rabin olp israelenses
0.43 - eleitoral eleições eleição candidato candidatos votos campanha voto turno partidos

I.4 Tópicos mais bem avaliados pelos vetores Skip-Gram - Wikipedia

0.60 - dia dias mês julho maio março abril dezembro agosto junho
0.55 - josé carlos silva luiz joão roberto pereira lima alves santos
0.55 - partido candidato eleitoral eleições eleição psdb pmdb candidatos votos governador
0.53 - escolas escola educação alunos curso ensino professores aulas cursos grau
0.52 - tv rádio programa televisão globo rede canal programas emissoras emissora
0.50 - ônibus transporte metrô paulo trem transportes linha estação passageiros trens
0.49 - saúde hospital médico médicos atendimento hospitais medicina médica jatene
0.48 - importação produtos indústria governo comércio importações exportações setor
0.47 - maria ribeiro joão ana jorge lima alves josé rosa pedro
0.47 - restaurante cozinha pratos comida casa restaurantes carne comer cardápio prato
0.47 - câmara deputado comissão deputados senado pmdb congresso projeto parlamentares
0.47 - música orquestra ópera compositor concerto piano musical maestro músicos teatro
0.47 - carro carros veículos fábrica ford veículo fiat motor modelo volkswagen
0.47 - fazer mim sempre porque nunca acho sei quero vou coisa
0.47 - poeta poesia literatura obra escritor autor livro poema poemas crítico
0.46 - cor cores olhos azul cabelo branco pele cabelos vermelho verde
0.46 - feira semana segunda sexta última quarta quinta terça passada próxima
0.45 - estados norte unidos eua américa brasil americano países americanos americana
0.45 - militar militares guerra exército general regime forças armadas durante oficiais
0.45 - livro livros escritor autor editora romance literatura sobre história obra